ANALIZA DANYCH

Wykład 2: Przygotowanie danych



dr inż. Bartosz Czech

Katedra Genetyki · Uniwersytet Przyrodniczy we Wrocławiu

bartosz.czech@upwr.edu.pl

Plan i cel

Plan wykładu

1. Znaczenie przygotowania danych
2. Podział i eksploracja danych
3. Wartości brakujące — mechanizmy i strategie
4. Wartości odstające, skale cech, kodowanie
5. Specyfika danych mikrobiomowych: zera, normalizacja, kompozycyjność, filtrowanie, CLR
6. Efekt partii (*batch effect*)
7. Przeciek (*data leakage*) i potok

Cel

Zrozumieć, co należy zrobić z danymi **przed** modelowaniem — i jak zrobić to **bez zaglądania do zbioru testowego**.

Przygotowanie danych — skala zadania

- W realnych projektach przygotowanie danych zajmuje **więcej czasu niż modelowanie** — i częściej decyduje o wyniku
- Model jest tak dobry, jak dane, które otrzymuje

Zasada „garbage in, garbage out”

Żaden algorytm nie naprawi źle przygotowanych danych. Jakość wejścia wyznacza górną granicę jakości modelu.

Dane demonstracyjne kursu

Wykłady opieramy na jednej kohorcie demonstracyjnej — metagenomach jelitowych pacjentów z rakiem jelita grubego i osób zdrowych (*ZellerG_2014*):

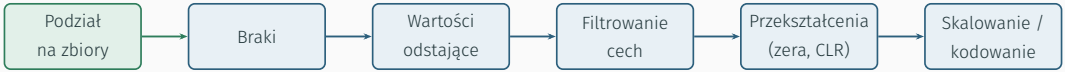
```
src <- file.path(path.expand("~/analiza_danych_bioinf/wyklady"), "load_cached.R")
source(src)
d <- load_demo()      # ZellerG_2014: kontrola i rak (gruczolaki pominięte)
dim(d$X)
table(d$y)
```

```
[1] 114 652

control     CRC
     61      53
```

- 114 próbek, 652 taksony — klasy niemal zrównoważone; na tych danych pokażemy dziś każdy krok obróbki
- `load_demo()` — funkcja pomocnicza kursu: `ZellerG_2014` (względne liczebności) z `curatedMetagenomicData`, kontrole i rak

Etapy przygotowania danych



- podział — **przed** wszystkimi krokami; kolejność dalszych kroków nie jest przypadkowa (np. filtr cech przed CLR)
- kroki z parametrami (mediana, próg, średnia) wyznaczają je **na zbiorze treningowym**

Katalog typowych problemów

Problemy ogólne

- wartości brakujące (NA)
- wartości odstające
- różne skale cech
- cechy o znikomej wariancji
- kategorie do zakodowania

Specyfika danych biologicznych

- rzadkość (wiele zer)
- kompozycyjność
- $p \gg n$
- efekt partii

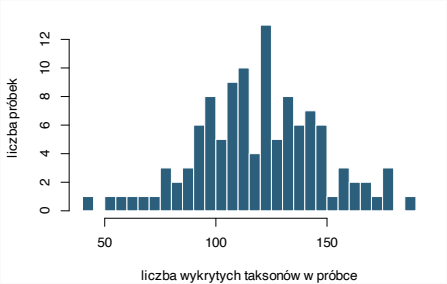
Pierwszy przegląd liczbowy

Przed czyszczeniem danych warto obliczyć podstawowe statystyki surowej macierzy cech:

```
round(100 * mean(as.matrix(d$X) == 0), 1)  # % zer
summary(rowSums(d$X > 0))   # taksonów na próbkę
```

```
[1] 81.7
```

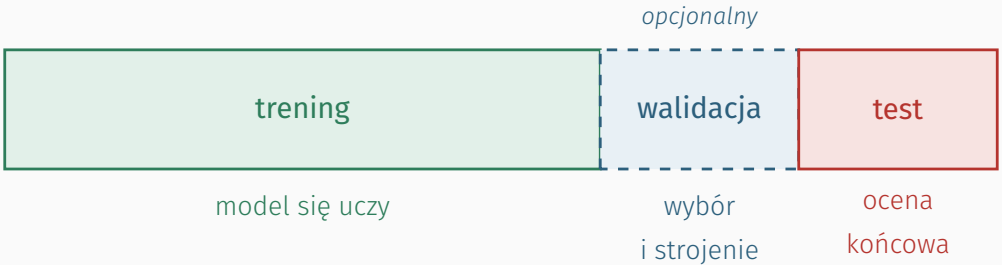
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   40.0   101.0   121.0   119.6   139.8   187.0
```



82% macierzy to zera; typowa próbka zawiera ok. **120** z 652 taksonów, a liczba wykrytych taksonów waha się blisko pięciokrotnie (40–187). To nie są braki pomiaru, lecz niewykrycia taksonu (zero a NA — w części o mikrobiomie). Opis całej kohorty — bez wpływu na decyzje obróbki.

Podział i eksploracja danych

Podział danych — przypomnienie (W1)



- **trening** — model dopasowuje parametry (np. współczynniki)
- **walidacja** (opcjonalna) — wybór metody i hiperparametrów, czyli ustawień wybieranych przed uczeniem (np. siły kary); przy małych zbiorach zastępuje ją walidacja krzyżowa
- **test** — jedna, końcowa ocena; symuluje „nowych pacjentów”
- proporcje np. 60/20/20; u nas bez osobnej walidacji: 75/25 (86 i 28 próbek)

Podział warstwowy

Podział warstwowy (*stratified split*)

Losowanie zbioru testowego osobno w każdej klasie, tak aby proporcje klas w treningu i teście odpowiadały proporcjom w całości.

```r
set.seed(42)
idx_test <- unlist(lapply(split(seq_along(d$y), d$y),
                          function(i) sample(i, round(0.25 * length(i)))))
table(d$y[-idx_test])   # trening
table(d$y[idx_test])    # test
```

```
control     CRC
     46      40

control     CRC
     15      13
```

- trening 46/40, test 15/13 — proporcje jak w całości (61/53)
- `set.seed` czyni podział **powtarzalnym**

Cel eksploracji danych

- wychwycić problemy, zanim trafią do modelu (odstające, partie, braki)
- wstępnie ocenić, czy klasy są rozróżnialne
- dobrać obróbkę do *tych* danych, nie do podręcznikowego ideału

Zasada

Eksploracja, z której wynikają decyzje obróbki, prowadzona jest na **treningu**; decyzje oparte na całości danych — łącznie z testem — to już forma przecieku. Sam opis struktury całej kohorty, bez oceny modelu, może korzystać z całości.

Eksploracja — lista kontrolna

- rozkład każdej cechy: skośność, wartości skrajne, nietypowe skupienia wartości
- rozkład klas: czy klasy są zrównoważone (u nas 61 i 53 — tak)
- częstość występowania i rzadkość: ile cech warto odfiltrować
- rzut próbek na płaszczyznę (PCA, W3): skupienia, odstające, układ wg partii
- korelacje: cechy niemal identyczne, pozorne zależności kompozycyjne

> Każdy z tych punktów prowadzi do konkretnej decyzji w potoku obróbki — eksploracja nie jest „oglądaniem", lecz projektowaniem przygotowania danych.

Wartości brakujące

Źródła braków danych

- nieudany pomiar lub kontrola jakości odrzucająca odczyt
- brak danej w metadanych klinicznych (np. nieuzupełniony wiek)
- scalanie zbiorów o różnym zakresie zmiennych

Uwaga terminologiczna

W mikrobiomie **zero** (taksonu nie wykryto) to *nie* to samo co **NA** (brak pomiaru). Do tego rozróżnienia wrócimy w części o danych mikrobiomowych.

Trzy mechanizmy powstawania braków

Mechanizm	Charakterystyka (Rubin, 1976)
całkowicie losowy (MCAR, *missing completely at random*)	prawdopodobieństwo braku nie zależy od żadnych wartości, ani obserwowanych, ani brakujących
losowy warunkowo (MAR, *missing at random*)	prawdopodobieństwo braku zależy od wartości *obserwowanych*, lecz nie od samej brakującej wartości
nielosowy (MNAR, *missing not at random*)	prawdopodobieństwo braku zależy od *samej* brakującej wartości

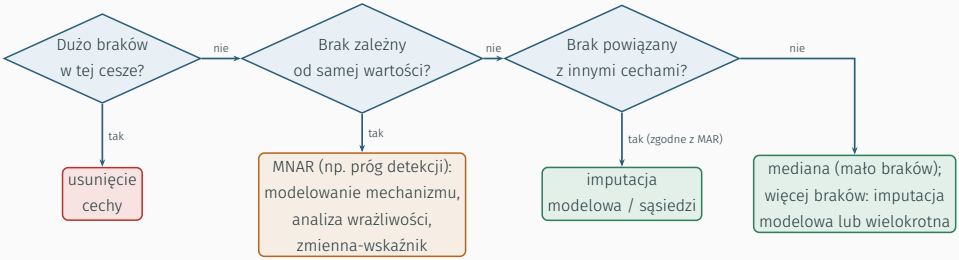
Mechanizm braku decyduje o tym, czy imputacja jest bezpieczna.

MCAR / MAR / MNAR — przykłady z badań mikrobiomowych

- **MCAR** — probówki uszkodzone w transporcie; to, których próbek dotyczy uszkodzenie, jest przypadkowe i niezwiązane z pacjentem
- **MAR** — starsi pacjenci rzadziej mają uzupełniony pomiar BMI; brak zależy od *wieku*, który znamy — można go wykorzystać do imputacji
- **MNAR** — stężenie poniżej progu detekcji aparatu; brakuje właśnie *najniższych* wartości — brak niesie informację o samej wartości

MNAR jest najtrudniejszy do obsłużenia: usunięcie lub prosta imputacja takich braków **systematycznie zniekształca** rozkład cechy.

Postępowanie z brakami — drzewo decyzji



Reguła praktyczna: cecha w większości pusta — usunięcie; przesłanki (z wiedzy o pomiarze), że brak zależy od samej wartości — modelowanie mechanizmu braku; brak powiązany z innymi cechami — imputacja modelowa; w pozostałych przypadkach — mediana przy niewielu brakach.

Braki w naszych danych — realny przykład

Braki w metadanych klinicznych kohorty (n = 114; `fobt` — test na krew utajoną w kale):

```
labs <- c("BMI", "fobt", "ajcc", "tnm")
na_tab <- data.frame(variable = labs,
  n_na = sapply(d$cd[labs], function(z) sum(is.na(z))),
  pct = round(100 * sapply(d$cd[labs], function(z) mean(is.na(z))), 1),
  row.names = NULL)
print(na_tab, row.names = FALSE)
```

```
 variable n_na  pct
      BMI    4  3.5
     fobt    1  0.9
     ajcc   61 53.5
      tnm   61 53.5
```

- `ajcc`, `tnm` (stadium raka) — brak u **wszystkich 61** kontroli: brak **strukturalny**, cecha istnieje tylko u chorych — usunąć
- `BMI`, `fobt` — pojedyncze braki: możliwa imputacja

Strategie wobec braków

Strategia	Kiedy	Ryzyko
usunięcie próbek	braków niewiele, losowe	utrata danych
usunięcie cech	cecha w większości pusta	utrata sygnału
imputacja	braków umiarkowanie	wprowadzenie obciążenia

Nie istnieje jedna uniwersalna strategia — wybór zależy od skali i mechanizmu braków.

Imputacja — najczęstsze metody

Imputacja (*imputation*)

Uzupełnienie brakującej wartości oszacowaniem wyznaczonym z pozostałych danych.

- **stała** — np. zero lub wartość spoza zakresu (dla braków strukturalnych — o ile sam fakt braku nie ujawnia zmiennej celu, por. `ajcc/tnm`)
- **średnia / mediana** cechy — prosta, lecz zawęża zmienność
- **najbliżsi sąsiedzi** — średnia z kilku próbek najbardziej podobnych pod względem pozostałych cech (metoda k najbliższych sąsiadów — W4)
- **modelowa** — przewidywanie braku z pozostałych cech

Imputacja jedną wartością (średnią, medianą) **zaniża** wariancję cechy; imputacja modelowa może sztucznie **wzmacniać** zależności między cechami. Wartość uzupełniona to oszacowanie, nie pomiar.

Imputacja bez przecieku — demonstracja

Medianę liczymy **tylko na treningu** (podział `idx_test` z części o podziale danych) i tą samą wartością uzupełniamy test:

```
bmi <- d$cd$BMI                          # BMI, 4 braki
median(bmi[-idx_test], na.rm = TRUE)     # mediana treningowa
median(bmi, na.rm = TRUE)                # mediana z całości — PRZECIEK
```

```
[1] 24
[1] 25
```

> Parametr imputacji wolno estymować wyłącznie na treningu: mediana z całości (25) zawiera informację ze zbioru testowego. Wszystkie 4 braki BMI są tu w treningu; tę samą medianę (24) stosuje się do przyszłych braków w teście i u nowych pacjentów.

Wartości odstające, skale, kodowanie

Wartości odstające

Wartość odstająca (*outlier*)

Obserwacja wyraźnie odbiegająca od pozostałych, co budzi podejrzenie, że powstała w innym procesie niż reszta danych.

- mogą być błędem technicznym *albo* prawdziwym sygnałem biologicznym
- silnie wpływają na metody oparte na średniej i wariancji

Postępowanie

Najpierw **zrozumieć** źródło wartości odstającej, dopiero potem decydować: korekta, przekształcenie (np. logarytm) lub pozostawienie.

Wykrywanie wartości odstających

Pojedyncza cecha

- rozkład: histogram, wykres pudełkowy
- reguła oparta na rozstępie ćwiartkowym (IQR)
- wartości fizycznie niemożliwe (np. ujemne stężenie)

Wiele cech naraz

- cechy podsumowujące próbkę: głębokość sekwencjonowania (liczba odczytów), bogactwo (*richness* — liczba wykrytych taksonów)
- odległość profilu próbki od profilu przeciętnego
- rzut próbek na płaszczyznę (PCA, W3)

PCA (analiza głównych składowych) rzutuje próbki na kilka nowych osi, wzdłuż których dane różnią się najbardziej — setki cech da się wtedy obejrzeć na jednym wykresie płaskim. Metodę omówimy szczegółowo na wykładzie 3.

Reguła 1,5 IQR i wykres pudełkowy

Reguła 1,5 IQR (*Tukey's fences*)

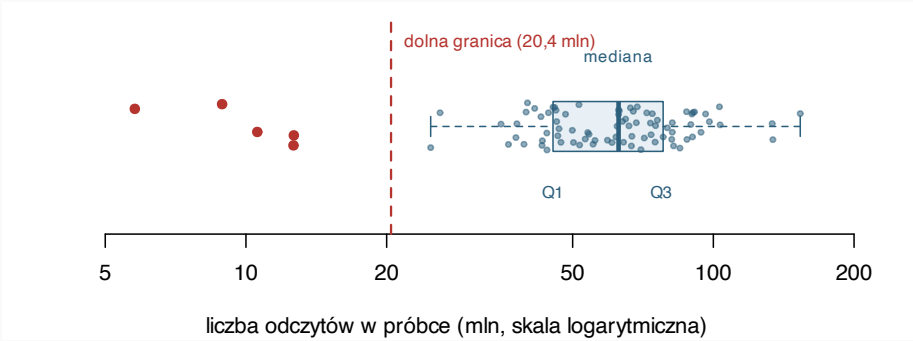
Wartość odstająca to wartość leżąca dalej niż półtora rozstępu międzykwartylowego poniżej dolnego kwartyla albo powyżej górnego kwartyla (Tukey, 1977).

- **kwartyl dolny (Q1)** i **górny (Q3)** — wartości, poniżej których leży odpowiednio 25% i 75% obserwacji
- **rozstęp międzykwartylowy (IQR)** — odległość od Q1 do Q3; obejmuje środkowe 50% obserwacji
- **wykres pudełkowy**: pudełko od Q1 do Q3, kreska w środku to mediana, wąsy sięgają do najdalszych wartości w granicach reguły, punkty poza wąsami — odstające
- dla cech silnie prawostronnie skośnych regułę warto stosować po logarytmowaniu

Głębokość sekwencjonowania w zbiorze treningowym

Głębokość sekwencjonowania (*sequencing depth*)
Liczba odczytów DNA uzyskanych z jednej próbki.

Bogactwo (*richness*)
Liczba taksonów wykrytych w jednej próbce.



- 81 z 86 próbek treningu: od 24,8 do 153,4 mln odczytów; mediana (wszystkich 86) 62,7 mln
- **5 próbek** (czerwone punkty) leży poniżej dolnej granicy reguły — 20,4 mln

Próbki o małej głębokości — wynik w R

```r
depth <- d$cd$number_reads[-idx_test] / 1e6   # odczyty (mln), trening
rich <- rowSums(d$X[-idx_test, ] > 0)         # bogactwo, trening
q <- quantile(log10(depth), c(0.25, 0.75))    # kwartyle na skali log
iqr <- q[2] - q[1]
granice <- unname(10^c(q[1] - 1.5 * iqr, q[2] + 1.5 * iqr))
out <- depth < granice[1] | depth > granice[2]
round(granice, 1)
data.frame(odczyty_mln = round(depth[out], 1), bogactwo = rich[out],
           grupa = d$y[-idx_test][out], row.names = NULL)
```

```
[1]  20.4 172.5
  odczyty_mln bogactwo   grupa
1        10.6       93     CRC
2         5.8       52     CRC
3        12.7       40     CRC
4        12.6       92 control
5         8.9       73 control
```

- 5 próbek ma 5–11 razy mniej odczytów niż mediana (62,7 mln)
- ich bogactwo (40–93) jest niższe od mediany treningu (121): przy płytkim sekwencjonowaniu rzadkie taksony nie zostają wykryte
- występują w obu grupach (3 CRC, 2 kontrole)
- to **sygnał do sprawdzenia**, nie automatyczny powód do usunięcia

Odstająca: artefakt czy sygnał?

Raczej artefakt	Raczej sygnał
bardzo niska głębokość sekwencjonowania	spójny, nietypowy profil biologiczny
próbka nie spełnia kryteriów kontroli jakości	powtarza się u podobnych pacjentów
pojedyncza, izolowana wartość	zgodna z opisem klinicznym

Kryterium usuwania próbek ustala się na treningu i stosuje jednakowo do nowych próbek — nigdy po obejrzeniu wyniku na teście.

Różne skale cech

Cechy w różnych jednostkach i zakresach nie są porównywalne wprost:

- cecha o dużym zakresie **zdominuje** obliczenia odległości
- kara za wielkość współczynników (regularyzacja) zależy od jednostek cechy

Przykład

Jeśli jedna cecha ma zakres 0–1, a druga 0–$10\,000$, to w odległości euklidesowej wpływ pierwszej jest pomijalny — niezależnie od jej znaczenia biologicznego.

Standaryzacja i normalizacja

Standaryzacja (z-score)

Od każdej wartości odejmujemy średnią cechy i dzielimy przez jej odchylenie standardowe. Cecha ma wtedy średnią 0 i odchylenie 1.

Normalizacja min–max

Każdą wartość przeskalowujemy liniowo tak, by najmniejsza trafiła na 0, a największa na 1 — cecha mieści się wtedy w przedziale od 0 do 1.

- obie sprowadzają cechy do porównywalnej skali
- obie są wrażliwe na wartości odstające (min–max najsilniej); wariant odporny używa mediany i rozstępu międzykwartylowego (IQR)
- **parametry** (średnią, odchylenie, wartości skrajne) estymujemy **na treningu**

Skalowanie — zakres stosowania

Standaryzacja jest wymagana dla części metod, dla innych jest zbędna:

Metoda	Skalować?	Dlaczego
k-średnich, kNN, SVM	tak	opierają się na odległościach
regresja z regularyzacją	tak	kara zależy od wielkości współczynników, a ta — od jednostek cechy
PCA na cechach w różnych jednostkach	tak	kieruje się wariancją
PCA na danych CLR (wspólna skala)	zwykle nie	cechy mają wspólną jednostkę
drzewa, lasy losowe, boosting	nie	podział zależy tylko od uporządkowania wartości

Reguła praktyczna: skalować, gdy metoda porównuje cechy ze sobą (odległości, wariancja, kara za współczynniki). Metody z tabeli omówimy w W3–W6.

Kodowanie zmiennych kategorycznych

Większość modeli przyjmuje wyłącznie liczby — kategorie trzeba zakodować:

- **kodowanie zero-jedynkowe** (*one-hot*) — po jednej kolumnie na kategorię, wartości 0/1
- unikać kodowania „porządkiem liczbowym” tam, gdzie kategorie nie mają porządku (sugeruje nieistniejącą kolejność)

Uwaga

Zbiór kategorii również ustalamy na treningu. Kategoria, która pojawi się tylko w teście, wymaga świadomej obsługi.

Specyfika danych mikrobiomowych

Brak danych a wartość zero

W danych mikrobiomowych należy rozróżnić dwie sytuacje:

Zero

Taksonu *nie wykryto* — wartość jest znana i równa zeru (brak taksonu albo zbyt mała głębokość). Zera nie usuwa się ani nie uzupełnia jak braku.

NA

Pomiaru *brak* — to luka w danych. NA trzeba świadomie obsłużyć.

Uwaga

Zer i braków **nie należy** utożsamiać. Potraktowanie zer jako braków (lub odwrotnie) zniekształca dane, zanim jeszcze zaczniemy modelować.

Zera w danych mikrobiomowych — przyczyny

Zero w tabeli liczebności może oznaczać trzy różne rzeczy:

Rodzaj zera	Przyczyna
strukturalne	takson nie występuje u tego gospodarza / w tym środowisku
próbkowania	takson jest, ale rzadki — nie trafił do próbki
detekcji	takson jest, lecz poniżej progu wykrywalności metody

> Dwa z trzech rodzajów zer mają przyczynę **techniczną**, a z samej wartości zero nie da się odczytać, która z przyczyn zadziałała.

Normalizacja próbek względem głębokości sekwencjonowania

Różne próbki mają różną głębokość sekwencjonowania — surowych liczebności nie można porównywać wprost.

Podejście	Co robi	Uwagi
udziały (TSS, *total sum scaling*)	dzieli liczbę odczytów taksonu przez sumę odczytów próbki	suma stała staje się jawna
rarefakcja	podpróbkowuje do wspólnej głębokości	wymaga liczb zliczeń
przekształcenia log-ratio	porównują taksony z sobą w obrębie próbki (dalej)	zalecane dla ML

> Nie mylić z normalizacją min–max cech: tu przelicza się *każdą próbkę* (wiersz) osobno, tam — *każdą cechę* (kolumnę) na podstawie treningu. Nasze dane są już **udziałami** w procentach (suma 100 na próbkę).

Rarefakcja — idea i jej krytyka

Rarefakcja (*rarefaction*)

Losowanie z każdej próbki tej samej liczby odczytów (bez zwracania), aby wyrównać głębokość sekwencjonowania przed porównaniem próbek, np. ich bogactwa.

- **wada:** odrzuca część zebranych danych i dodaje losowy szum

Rarefakcja bywa krytykowana jako statystycznie nieefektywna (McMurdie & Holmes, 2014). Wymaga też *liczb zliczeń*, których nasze względne liczebności nie zawierają — dlatego w tym potoku jej nie stosujemy.

Kompozycyjność

Dane kompozycyjne (*compositional data*)

Wektory dodatnich składowych o stałej sumie, niosące informację wyłącznie względną, czyli o udziałach części w całości (Aitchison, 1986).

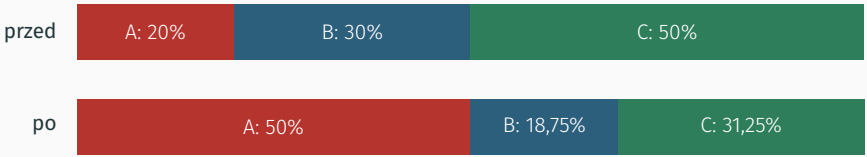
Udziały taksonów (u nas w procentach) sumują się do 100.

- wzrost udziału jednego taksonu **wymusza** spadek udziałów pozostałych
- surowe proporcje są mylące dla metod zakładających niezależność cech
- pozorna korelacja między taksonami może wynikać z samej sumy stałej

Wniosek

Danych kompozycyjnych nie analizuje się jak zwykłych liczb — stosuje się przekształcenia log-ratio, które zamieniają udziały na ilorazy i pozwalają korzystać ze standardowych metod.

Suma stała w praktyce



- Wzrost udziału A wymusza spadek B i C przy stałej liczebności — zależność pozorna.

Pozorna korelacja — symulacja

Dwa **niezależne** taksony i jeden dominujący; liczebności bezwzględne a udziały:

```
set.seed(1)
n <- 100
a <- rlnorm(n, 0, 0.5)      # takson A: liczebność bezwzględna
b <- rlnorm(n, 0, 0.5)      # takson B: niezależny od A
dom <- rlnorm(n, 2, 1)      # takson dominujący, silnie zmienny
total <- a + b + dom
round(c(absolute = cor(a, b), shares = cor(a / total, b / total)), 3)
```

```
absolute   shares
  -0.051    0.595
```

> Liczebności A i B nie są skorelowane, a ich **udziały** — wyraźnie tak: oba dzieli się przez tę samą, zmienną sumę. To klasyczna korelacja pozorna (Pearson, 1897).

Cechy o znikomej wariancji

- cecha **stała** nie niesie informacji i nie może zostać wykorzystana przez model
- cecha o **znikomej** wariancji zwykle wnosi głównie szum

Kryterium bliskiej zeru wariancji (*near-zero variance*)

Cecha jest podejrzana, gdy **jednocześnie**: ma bardzo mało unikalnych wartości *oraz* jedna wartość silnie dominuje (wysoki stosunek częstości najczęstszej do drugiej w kolejności).

Rzadkość i filtrowanie cech

Przy $p \gg n$ warto usunąć cechy wnoszące przede wszystkim szum:

- taksony obecne u **niewielkiej liczby** pacjentów (niska częstość występowania)
- cechy o **zerowej wariancji** (stałe — bezużyteczne dla modelu)

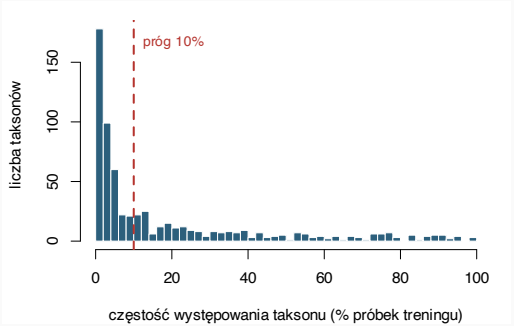
Filtr to również krok uczący się z danych

To, które taksony przechodzą próg, ustala się na zbiorze treningowym, nie na całości danych — w przeciwnym razie test współdecyduje o zestawie cech (przeciek).

Filtr częstości występowania w liczbach

```r
prev <- colMeans(d$X[-idx_test, ] > 0)
thr <- c(.01, .05, .10, .25, .50)
print(data.frame(threshold = thr,
  taxa = sapply(thr, function(t)
    sum(prev >= t))), row.names = FALSE)
```

```
 threshold taxa
      0.01  602
      0.05  342
      0.10  272
      0.25  166
      0.50   89
```



Próg kursu (częstość występowania $\geq$ 10% w treningu) zostawia **272** z 652 taksonów; pozostałe 380 (58%) wykryto u mniej niż 10% pacjentów, w tym 128 tylko u jednego, a 50 — u żadnego.

$p \gg n$ i selekcja cech

Więcej cech niż próbek to główne źródło przeuczenia. Trzy podejścia:

Podejście	Idea
filtrowanie	odsianie cech słabych niezależnie od modelu (wariancja, częstość występowania)
regularyzacja	kara za duże współczynniki wbudowana w model (np. LASSO)
redukcja wymiarów	zastąpienie p cech kilkoma składowymi (np. PCA, W3)

Zastępowanie zer

Przekształcenia log-ratio (następny slajd) wymagają logarytmu, a $\log 0$ jest nieokreślony. Przed nimi zera zastępuje się małą wartością dodatnią:

- **stała pseudoliczność** (*pseudocount*) — dodanie małej stałej do każdej komórki (w kursie 10^{-6}); nazwa od dodawania jednego „sztucznego” odczytu do zliczeń; proste, lecz arbitralne
- **zastąpienie multiplikatywne** — np. `zCompositions::multRepl` dla udziałów (`cmultRepl` dla zliczeń): zero zastępuje się wartością poniżej progu wykrycia, a pozostałe udziały proporcjonalnie zmniejsza

Zastąpienie zer a przeciek

Stała pseudoliczność działa w każdej próbce osobno — nie powoduje przecieku. Metody szacujące wartość zastępczą z innych próbek dopasowuje się **na treningu**.

Przekształcenie log-ratio (CLR)

Przekształcenie CLR (*centered log-ratio*)

Logarytm ilorazu udziału taksonu do średniej geometrycznej udziałów wszystkich taksonów **tej samej** próbki.

- **średnia geometryczna** — średnia w skali logarytmicznej: logarytmuje się wartości, uśrednia je i wraca do skali pierwotnej
- CLR > 0: takson obfitszy niż przeciętny takson próbki; CLR < 0: rzadszy
- liczone w **każdej próbce osobno** — nie korzysta z innych próbek
- wartości CLR w próbce sumują się do zera: ograniczenie sumy stałej nie znika, lecz przyjmuje postać wygodną dla standardowych metod

CLR na naszych danych — przed i po

Trening (86 próbek), 272 taksony z filtra treningowego:

```
keep <- colMeans(d$X[-idx_test, ] > 0) >= 0.10   # filtr: trening
Xf <- d$X[-idx_test, keep]                        # udziały (%)
Xclr <- clr(Xf)                                   # pseudoliczność 1e-6
round(quantile(as.matrix(Xf), c(0, .5, 1)), 2)
round(quantile(Xclr, c(0, .5, 1)), 2)
```

```
   0%   50%  100%
 0.00  0.00 62.33
   0%   50%  100%
-6.80 -3.37 15.15
```

- surowe udziały: większość to zera, kilka taksonów dominuje (do 62%)
- po CLR: wartości ujemne dla taksonów rzadszych niż przeciętny takson próbki

Korelacja taksonów a skala analizy

Dwa najobfitsze taksony treningu; korelacja na surowych udziałach i po CLR (`Xf`, `Xclr` z poprzedniego slajdu):

```
ab <- order(colMeans(Xf), decreasing = TRUE)[1:2]
sub("species:", "", colnames(Xf)[ab])
round(c(shares = cor(Xf[, ab[1]], Xf[, ab[2]]),
        clr = cor(Xclr[, ab[1]], Xclr[, ab[2]])), 3)
```

```
[1] "Ruminococcus bromii"  "Phocaeicola vulgatus"
shares    clr
-0.174 -0.088
```

> Korelacja **zależy od skali**, w której ją liczymy (**-0,17** na udziałach, **-0,09** po CLR). Żadna z tych liczb nie jest „prawdziwą” korelacją biologiczną — CLR też liczy się względem średniej próbki.

Efekt partii

Efekt partii — definicja i źródła

Efekt partii (*batch effect*)

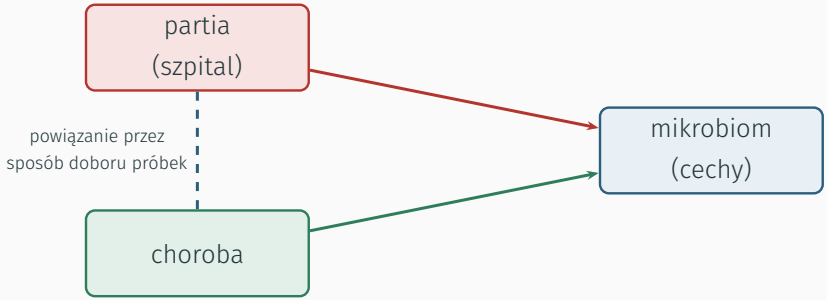
Systematyczna różnica w pomiarach między grupami próbek przetwarzanych w różnych warunkach technicznych (partiach), niezwiązana z badaną biologią.

Źródła różnic między partiami w metagenomice:

- protokół ekstrakcji DNA i przygotowania biblioteki
- platforma i seria sekwencjonowania, głębokość odczytu
- laboratorium, kraj, operator
- wersja potoku bioinformatycznego i bazy referencyjnej

> Przy łączeniu kohort z różnych badań efekt partii bywa **silniejszy niż sygnał chorobowy** — to jedna z głównych pułapek metagenomiki porównawczej.

Efekt partii jako czynnik zakłócający



Gdy chorzy pochodzą z jednej partii, a zdrowi z drugiej, model nauczy się **rozpoznawać partię**, a my odczytamy to jako „wykrycie choroby”. Partia jest tu **czynnikiem zakłócającym** (*confounder*): zmienną powiązaną jednocześnie z cechami i ze zmienną celu, która może wytworzyć pozorny związek.

Wykrywanie efektu partii

- **rzut próbek na płaszczyznę** (PCA, W3) pokolorowany zmienną techniczną — układ próbek według partii, a nie według choroby, wskazuje na efekt partii
- sprawdzić, czy zmienna celu nie jest powiązana z partią
- głębokość sekwencjonowania jako podejrzana zmienna towarzysząca (kowariata)

> Nawet w jednej kohorcie głębokość sekwencjonowania waha się **ponad 26-krotnie** (w treningu od 5,8 do 153,4 mln odczytów; zob. wykres pudełkowy w części o wartościach odstających) — to realne źródło zmienności technicznej, które warto kontrolować.

Korekcja efektu partii — ograniczenia

- metody typu `ComBat` (pakiet `sva`) modelują i odejmują różnice między partiami
- wymagają **znajomości** przypisania próbek do partii

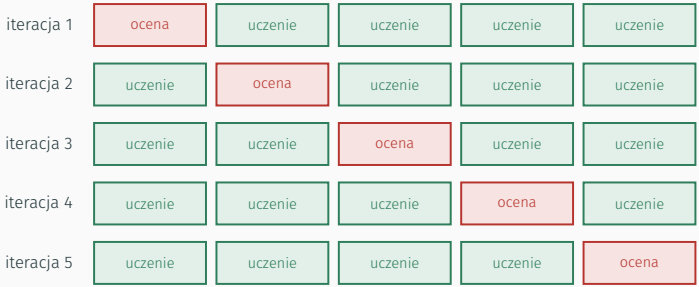
Pułapka nadkorekcji

Jeśli partia jest silnie powiązana ze zmienną celu, korekcja albo **usuwa sygnał**, albo — gdy uwzględni zmienną celu — może go **sztucznie wzmacniać**. Skuteczniejsze jest zapobieganie: zrównoważenie przypadków i kontroli w obrębie partii już na etapie projektu badania.

Przeciek i potok

Walidacja krzyżowa — przypomnienie (W1)

Zbiór treningowy dzielony na k równych części (ang. *fold*); model uczy się k razy — jedna część do oceny, pozostałe do uczenia; wynik to **średnia** z k przebiegów.

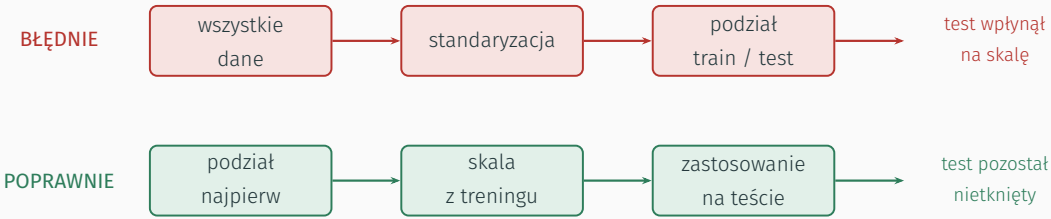


Kroki obróbki z parametrami (imputacja, filtr, skalowanie) dopasowuje się **w każdej iteracji od nowa** — tylko na częściach uczących.

Przeciek danych — przypomnienie (W1)

> Przeciek: użycie przy uczeniu lub doborze modelu informacji niedostępnej przy predykcji dla nowych danych, zwłaszcza informacji ze zbioru testowego; skutek — zawyżona ocena modelu.

Przykład: standaryzacja policzona na **całym** zbiorze przed podziałem miesza informację z testu do treningu:



- tak samo: imputacja i selekcja cech na całości danych, strojenie modelu na teście
- osobny przypadek: **duplikaty** tej samej próbki po obu stronach podziału

Przeciek w mikrobiomie — typowe źródła

Specyfika danych mikrobiomowych daje własne, łatwe do przeoczenia źródła przecieku:

- **filtr częstości występowania** dobrany na całości — test współdecyduje, które taksony zostają, a więc także względem czego liczona jest średnia geometryczna w CLR (u nas 262 zamiast 272 taksonów)
- **zastępowanie zer** metodą korzystającą z innych próbek oraz **korekcja partii** oszacowane poza treningiem

Samo CLR (ze stałą pseudolicznością) liczy się w każdej próbce osobno — **nie** jest źródłem przecieku.

Kroki, które *uczą się* czegoś z innych próbek, mogą korzystać wyłącznie ze zbioru treningowego. Dlatego zamyka się je w potoku.

Potok jako ochrona przed przeciekiem

Potok (*pipeline*)

Wszystkie kroki przygotowania (imputacja, filtrowanie, skalowanie) zapisujemy jako **jeden obiekt**, który uczy swoich parametrów **tylko na treningu**, a następnie stosuje je bez zmian na teście.

Korzyści:

- ochrona przed przeciekiem działa automatycznie — mniejsze ryzyko błędu
- powtarzalność — ten sam potok działa na nowych danych
- czytelność — cała obróbka w jednym miejscu

Potok w R — recipes

```
library(recipes)
keep <- colMeans(d$X[-idx_test, ] > 0) >= 0.10   # filtr: trening
Z <- clr(d$X[, keep])                             # CLR: każda próbka osobno
dat <- data.frame(group = d$y, BMI = d$cd$BMI, age = d$cd$age, Z)
train <- dat[-idx_test, ]
test <- dat[idx_test, ]

rec <- recipe(group ~ ., data = train)
rec <- step_impute_median(rec, BMI)            # mediana BMI: z treningu
rec <- step_normalize(rec, BMI, age)           # średnia i SD: z treningu
rec_trained <- prep(rec, training = train)     # parametry liczone na treningu
test_prep <- bake(rec_trained, new_data = test)  # te same parametry na teście
```

- `prep()` uczy parametry na treningu, `bake()` stosuje je na teście — jak `fit`/`transform` w Pythonie
- `recipes` (tidymodels) — przykład spoza kursu; na ćwiczeniach ta sama zasada w base R

Co odczytujemy z potoku

```
rec_trained
dim(test_prep)
```

```
-- Recipe ----------------------------------------

-- Inputs
Number of variables by role
outcome:     1
predictor: 274

-- Training information
Training data contained 86 data points and 4 incomplete rows.

-- Operations
* Median imputation for: BMI | Trained
* Centering and scaling for: BMI age | Trained
[1]  28 275
```

- 274 predyktory = 272 taksony (CLR) + BMI + wiek
- 4 niepełne wiersze to braki BMI
- znacznik `Trained`: parametry z 86 próbek treningu
- test po `bake()`: 28 próbek, 274 predyktory + cel

Ta sama zasada w Pythonie

```
from sklearn.compose import ColumnTransformer
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

clinical = Pipeline([("impute", SimpleImputer(strategy="median")),
                     ("scale", StandardScaler())])
prep = ColumnTransformer([("clinical", clinical, ["BMI", "age"])],
                         remainder="passthrough")  # taksony (CLR) bez zmian
prep.fit(X_train)                    # parametry uczone TYLKO na treningu
X_test_p = prep.transform(X_test)    # te same parametry na teście
```

Inne nazwy, **ta sama zasada**: `fit` na treningu, `transform` na teście.

Podsumowanie

Do zapamiętania

- przygotowanie danych często decyduje o wyniku bardziej niż wybór modelu
- braki: rozpoznać mechanizm (MCAR/MAR/MNAR); imputację uczyć **na treningu**
- odstające: najpierw zrozumieć źródło, potem decydować
- metody oparte na odległości wymagają skalowania; drzewa decyzyjne go nie wymagają
- w mikrobiomie: **zero** $\neq$ **NA**; uwzględniać kompozycyjność (CLR), rzadkość i efekt partii
- **każdy** krok uczący się z danych dopasowuje się na treningu; potok chroni przed przeciekiem

Lista kontrolna przygotowania danych

1. podział danych (warstwowy, z ustalonym ziarnem) — *zanim* cokolwiek zostanie policzone
2. eksploracja zbioru treningowego: rozkłady, braki, częstość występowania, partie
3. decyzje o brakach i wartościach odstających (według mechanizmu, nie automatycznie)
4. filtr rzadkich taksonów (na treningu), zastąpienie zer, CLR
5. zamknięcie wszystkich kroków w potoku uczonym na treningu
6. zbiór testowy przeznaczony do **jednej**, końcowej oceny

Literatura

- **Boehmke & Greenwell**, *Hands-On Machine Learning with R* — inżynieria cech, near-zero variance, przygotowanie danych (rozdz. 3)
  https://bradleyboehmke.github.io/HOML/engineering.html
- **Kuhn & Silge**, *Tidy Modeling with R* — `recipes`: potoki przetwarzania bez przecieku (tidymodels, poza kursem)
  https://www.tmwr.org/
- **Gloor i in.** (2017), *Microbiome datasets are compositional* — CLR i analiza log-ratio w mikrobiomie
  Frontiers in Microbiology 8:2224
- **McMurdie & Holmes** (2014), *Waste not, want not* — krytyka rarefakcji
  PLoS Comput Biol 10(4):e1003531

Na następny wykład

Wykład 3: Uczenie nienadzorowane

- uczenie bez etykiet i jego różnice wobec nadzorowanego
- analiza głównych składowych (PCA): macierz kowariancji, wartości własne, wariancja wyjaśniona
- ordynacja i miary odległości między próbkami
- analiza skupień (k-średnich, metoda hierarchiczna) i ocena wyniku bez etykiet

Dziękuję za uwagę

dr inż. Bartosz Czech · bartosz.czech@upwr.edu.pl