Analiza danych z międzynarodowych badań edukacyjnych w R z użyciem pakietu Rrepest

Jakub Łobaszewski (j.lobaszewski@ibe.edu.pl)
Mateusz Kleczaj (m.kleczaj@ibe.edu.pl)

Spis treści

1 Wprowadzenie

Pakiet **Rrepest** w R (Ilizaliturri, Avvisati, & Keslair, 2025) jest rozwinięciem podejścia znanego z modułu **repest** w Stata, opracowanego pierwotnie dla analityków Organizacji Współpracy Gospodarczej i Rozwoju (OECD) (Avvisati & Keslair, 2014). Został zaprojektowany z myślą o ułatwieniu analizy danych z międzynarodowych badań edukacyjnych prowadzonych przez OECD oraz Międzynarodowe Stowarzyszenie Mierzenia Osiągnięć Szkolnych (IEA), takich jak:

- **PISA**, **PIAAC**, **TALIS**, **SSES** (badania OECD),
- **TIMSS**, **PIRLS**, **ICCS**, **ICILS** (badania IEA).

Pakiet umożliwia poprawną analizę wyników zgodnie z metodologią stosowaną w wymienionych wyżej badaniach, wspierając m.in.:

- obliczanie średnich, odsetków oraz częstości grupowanych,
- analizę korelacji i kowariancji oraz regresji liniowych,
- testy porównawcze między grupami oraz obliczanie miar rozproszenia.

Rrepest automatyzuje złożone elementy analizy danych z międzynarodowych badań edukacyjnych, uwzględniając stosowane w nich złożone schematy doboru próby oraz techniki skalowania wyników. Pakiet uwzględnia w obliczeniach **wagi statystyczne, wagi replikacyjne oraz wartości prawdopodobne (plausible values, PV)**, zapewniając prawidłowe oszacowanie niepewności pomiarowej. Dzięki temu pakiet ułatwia pracę z dużymi zbiorami danych i pozwala na szybsze dokonanie analiz zgodnych ze standardami OECD i IEA.

> **i** Dodatkowe zasoby
>
> - **Pełna dokumentacja** - plik PDF zawierający opis funkcji pakietu oraz praktyczne przykłady jego zastosowania w analizie danych z międzynarodowych badań edukacyjnych.
> - **README na CRAN** - skrócony przewodnik instalacyjny i opis głównych funkcji pakietu, dostępny na stronie pakietu w CRAN.
> - **Repo/wiki z przykładami (OECD GitLab)** - strona, na której znajdziemy aktualizacje pakietu publikowane przez OECD oraz dodatkowe przykłady analiz.

1.1 Dlaczego warto używać `Rrepest`?

Zalety pakietu `Rrepest`

- **Automatyzacja**: upraszcza przetwarzanie złożonych danych, automatycznie uwzględniając metodologię badań OECD i IEA, taką jak wagi replikacyjne i wartości prawdopodobne (PV). Eliminuje to konieczność samodzielnej implementacji skomplikowanych formuł metodologicznych.
- **Wygoda**: funkcja `Rrepest()` oferuje przejrzystą składnię do obliczania statystyk opisowych, korelacji, regresji liniowej, testów różnic oraz generowania tabel wynikowych.
- **Elastyczność**: umożliwia analizy danych z wielu międzynarodowych badań edukacyjnych.
- **Wsparcie**: pakiet jest aktywnie wykorzystywany i stale rozwijany przez OECD.
- **Szybkość**: wykorzystuje obliczenia równoległe, rozdzielając obciążenie na wiele rdzeni procesora, co pozwala na dwukrotnie szybsze przetwarzanie danych w porównaniu do alternatywnych rozwiązań, szczególnie przy dużych zbiorach danych.

1.2 Ograniczenia `Rrepest`

Ważne ograniczenia pakietu `Rrepest`

- Pakiet wspiera jedynie podstawowe typy analiz: średnie, rozkłady częstości, testy różnic, korelacje i regresje liniowe i logistyczne. Nie obsługuje np. obliczania poziomów umiejętności (benchmarków), percentyli oraz bardziej zaawansowanych analiz (np. modeli wielopoziomowych).
- Regresja logistyczna wykonywana za pomocą parametru **`est("log", ...)`** jest wspierana tylko dla badań OECD (np. PISA, TALIS).
- W regresji liniowej przeprowadzanej za pomocą funkcji `Rrepest()` wszystkie predyktory są traktowane jako zmienne ciągłe – również wtedy, gdy zostały zakodowane jako czynniki (factor).
- Jeśli w danych pozostaną etykiety SPSS (value labels), nazwy kolumn w tabelach wynikowych będą zawierały etykiety kategorii i zostaną posortowane alfabetycznie co może utrudniać czytelność wyników.

1.3 Obsługiwane badania

Badanie (Organizacja)	Obsługiwane przez `Rrepest`
PISA (OECD)	✓
PIAAC (OECD)	✓
TALIS (OECD)	✓
SSES (OECD)	✓
TIMSS (IEA)	✓
PIRLS (IEA)	✓
ICCS (IEA)	✓
ICILS (IEA)	✓

---

2 Instalacja pakietów i przygotowanie danych do analizy

Aby rozpocząć pracę, należy jednorazowo zainstalować pakiet:

```
# instalacja pakietu Rrepest:
install.packages("Rrepest")

# instalacja dodatkowych pakietów do pracy z danymi
install.packages(c("haven", "tidyverse"))

# haven: pakiet do obsługi w R plików .sav, .sas7bdat, .dta
# tidyverse: zbiór pakietów ułatwiających analizę danych, m.in. dplyr, purrr,
# readr, stringr
```

Poniżej prezentujemy kilka dobrych praktyk ułatwiających korzystanie z pakietu.

Dobre praktyki przy pracy z pakietem `Rrepest`:

- **Ujednolicenie pisowni nazw zmiennych**: Zamień wielkość liter w nazwach zmiennych na małe (np. `names(df) <- tolower(names(df))`), aby uniknąć problemów z odczytywaniem nazw przez funkcję `Rrepest()`. Niektóre jej argumenty (np. `over`) są wrażliwe na wielkość liter w podanych jako wartość nazwach zmiennych.
- **Sprawdzenie danych**: zweryfikuj strukturę zbioru i obecność braków danych:
  - struktura danych: `dplyr::glimpse(df)`, `summary(df)`,
  - liczba obserwacji w kategoriach danej zmiennej: `dplyr::count(df, zmienna)`,
  - liczba braków danych w kolumnach: (np. `colSums(is.na(df))`).
- **Powtarzalność**: Przechowuj kod w skryptach, dokumentuj ścieżki i transformacje, aby zapewnić powtarzalność analiz
- **Porównanie**: Porównaj wyniki analiz z oficjalnymi raportami OECD/IEA lub narzędziem **IEA IDB Analyzer**, aby upewnić się, że są zgodne z metodologią badania.

3 Składnia i kluczowe opcje funkcji `Rrepest()`

Podstawowa składnia funkcji wraz z jej głównymi argumentami

```
Rrepest(
  data = <zbiór danych>,
  svy  = <nazwa badania ("ALL"|"IALS"|"IELS"|"PIAAC"|"PIRLS"|"PISA"|
  "PISAOOS"|"SSES"|"TALIS" ("TALISSCH"/"TALISTCH")|"TALIS3S" ("TALISEC_STAFF"/
  "TALISEC_LEADER")"|"TIMSS"| "ICCS" (ICCS_T)"| "ICILS" | "SVY" (własne/
  niestandardowe badanie))>,
  est  = est(
    <"mean"|"var"|"std"|"quant"|"iqr"|"freq"|"corr"|"lm"|"log"|"cov"|"gen">,
    target = <zmienna>,
    regressor = <zmienna | wektor zmiennych>
    # argument "regressor" podajemy dla estymatorów"lm" i "log";
    # przy użyciu estymatora "gen" podstawiamy własny model, bez konieczności
    # używania argumentów "target" i "regressor"
  ),
  by   = <zmienna | wektor zmiennych>,
  over = <zmienna | wektor zmiennych>,
  test = TRUE/FALSE,
  n.pvs      = <liczba wartości prawdopodobnych (PV)>,
  cm.weights = c(<waga główna>, <wagi replikacyjne_1:K>),
  var.factor = <współczynnik wariancji>,
  show_na    = TRUE/FALSE
)
```

Najważniejsze argumenty funkcji:

- `svy` - nazwa badania, które bierzemy pod uwagę w analizie.
- `est()` – wbudowane estymatory to `"mean"`, `"var"`, `"std"`, `"quant"`, `"iqr"`, `"freq"`, `"corr"`, `"lm"`, `"log"`,`"cov"`, oraz ogólny `"gen"` do uruchamiania własnych modeli z R.

- `target` – pojedyncza zmienna lub wielokrotne imputacje PV przez zastosowanie składni z symbolem `@` (np. `PVREAD@`, `PVNUM@`, `ASMMATO@`). Składnia z symbolem `@` może być też używana przy podawaniu zmiennych PV w parametrach `by`, `over` i `regressor`.
- `by` – osobne estymacje/tabele dla wskazanych zmiennych.
- `over` + `test=TRUE` – analiza grup łącznie i raport testów różnic między poziomami zmiennej wskazanej w argumencie `over`.
- `n.pvs` – liczba PV wykorzystywana przy uśrednianiu wyników dla wartości prawdopodobnych (domyślnie `Rrepest` bierze pod uwagę 5 PV).

- `cm.weights` – główna waga i komplet wag replikacyjnych (np. `totwgt`, `jr1:jrK`). Ustawienie tego parametru jest konieczne przy niektórych analizach, w których nie korzystamy z wag domyślnie uwzględnianych przez funkcję `Rrepest()`.
- `var.factor` – współczynnik wariancji zgodny z zastosowanym schematem doboru próby. Ustawienie tego parametru jest konieczne przy niektórych analizach, w których nie korzystamy z wag domyślnie uwzględnianych przez funkcję `Rrepest()`.
- `show_na` – w przypadku ustawienia wartości tego parametru na "TRUE" w raportach częstości braki danych uwzględniane są jako osobna kategoria.

i `by` vs `over`

- `by` – wykonuje analizę osobno dla każdej grupy, traktując je jako niezależne zbiory danych. **Nie umożliwia bezpośredniego testowania różnic między grupami**
- `over` **w połączeniu z** `test=TRUE` – analizuje wszystkie grupy łącznie, automatycznie obliczając **różnice między nimi** i błędy standardowe.

3.1 Struktura danych w międzynarodowych badaniach edukacyjnych

Aby Rrepest działał poprawnie, kluczowe zmienne w zbiorze danych takie jak PV oraz wagi analityczne muszą mieć określoną strukturę nazw, odpowiadającą konwencjom w poszczególnych badaniach w obrębie danego cyklu badawczego.

3.1.1 Nazwy zmiennych z wartościami prawdopodobnymi (PV)

Wartości prawdopodobne (PV) to zestaw wielokrotnych imputacji niewidocznych (niedających się zmierzyć bezpośrednio) cech (np. umiejętności matematycznych), które pozwalają na lepsze oszacowanie błędów pomiaru. Funkcja `Rrepest()` rozpoznaje te wartości przez składnię zawierającą symbol `@` w nazwie zmiennej (np. `pv@math`, `pv@read`, `pv@civ`) i zakłada, że dostępne są wielokrotne jej imputacje (np. pv1math,..., pvKmath, gdzie K to liczba PV w danym badaniu).

i Ustawienie liczby PV

- Domyślnie funkcja `Rrepest()` przyjmuje 5 jako liczbę dostępnych imputacji.
- W przypadku, gdy w analizowanym badaniu jest więcej wartości prawdopodobnych (w badaniu PISA w cyklach od 2015 roku liczba PV wynosi 10), należy dodatkowo zdefiniować poprawną liczbę imputacji za pomocą argumentu `n.pvs` (np. n.pvs = 10).

3.1.2 Nazwy zmiennych wag analitycznych

Wagi analityczne są drugim rodzajem kluczowych zmiennych w międzynarodowych badaniach edukacyjnych. Aby funkcja `Rrepest()` działała poprawnie, nazwy tych zmiennych muszą przyjąć postać odpowiednią dla badania wybranego do analizy. W każdym badaniu występuje zmienna `Wagi głównej (total/final weight)`, np. `w_fstuwt` oraz `wagi replikacyjne (replicate weights)`, np. `w_fsturwt1,..., w_fsturwt80`. Nazwy tych zmiennych oraz liczby replik są różne w poszczególnych badaniach. W większości przypadków funkcja `Rrepest()` pobiera je automatycznie ze zbioru danych na podstawie nazwy badania podanej w argumencie `svy`.

i Przykładowe domyślne ustawienia wag analitycznych w funkcji `Rrepest()`

- **PISA** – waga główna `w_fstuwt` oraz 80 replik `w_fsturwt1,..., w_fsturwt80` (wagi uczniowskie)
- **ICCS/ICILS** – waga główna `totwgts` oraz 75 replik `srwgt1,..., srwgt75` (wagi uczniowskie)

! Ważne

- W przypadku badań TIMSS i PIRLS (oraz innych badań wykorzystujących schemat ważenia Jackknife Repeated Replication), konieczne jest samodzielne przeliczenie wag replikacyjnych i dodanie ich do zbioru danych. Opisujemy jak to zrobić w części praktycznej poradnika.
- W niektórych badaniach rozróżnia się schematy badań zależnie od analizowanych grup respondentów. Przykładowo, w badaniu TALIS w argumencie `'svy'` możemy podać `"TALISTCH"`, jeśli analizujemy nauczycieli, lub `"TALISSCH"`, jeśli dokonujemy analiz na zbiorze danych dotyczących szkół. Podobnie, w badaniu `ICCS` możemy podać w argumencie `'svy'` nazwę badania z przyrostkiem `'_T'` (`"ICCS_T"`), jeżeli analizujemy dane nauczycieli. Rrepest automatycznie zastosuje odpowiednie wagi analityczne.
- Metodologia badań TIMSS i PIRLS uległa zmianie w cyklach realizowanych po 2011 roku. W cyklach przeprowadzonych do 2011 roku włącznie stosowano 150 wag replikacyjnych oraz jedną wartość PV, natomiast w późniejszych cyklach – 250 wag replikacyjnych i pięć wartości PV.

4 Pobieranie danych

Dane i dokumentacja (np. opisy nazw zbiorów i zmiennych) z badań dostępne są na poniższych stronach:

- **PISA**: https://www.oecd.org/pisa/data/
- **TIMSS/PIRLS/ICCS/ICILS**: https://www.iea.nl/data-tools/repository
- **PIAAC**: https://www.oecd.org/skills/piaac/data/
- **TALIS**: https://www.oecd.org/en/about/programmes/talis.html#data
- **SSES**: https://www.oecd.org/en/about/programmes/oecd-survey-on-social-and-emotional-skills.html#data

W przypadku badań OECD przed pobraniem zbioru należy wypełnić krótką ankietę.

4.1 Struktura plików danych

Przed rozpoczęciem analizy dane z badań międzynarodowych muszą zostać zaimportowane do środowiska R, co wymaga zrozumienia złożonej struktury plików je zawierających. Struktura ta jest różna pomiędzy badaniami OECD i IEA.

4.1.1 Struktura danych IEA

Zbiory IEA są zazwyczaj podzielone na dużą liczbę plików, pogrupowanych według kraju, poziomu klasy oraz zastosowanego narzędzia badawczego. Przykładowo po ściągnięciu danych z badania TIMSS 2023 dla klasy 4 w folderze zobaczymy ponad 500 plików.

i Nazewnictwo plików w badaniach IEA, na przykładzie badania TIMSS

Przykładowo: Plik `asapolm8.sav` zawiera dane uczniów z Polski z badania TIMSS 2023 dla klasy 4 w formacie `.sav`
Gdzie:

- `asa`: odpowiedzi na zadania i wyniki uczniów klasy 4
- `pol`: kod kraju (Polska)
- `m8`: cykl badania (TIMSS 2023)

Pierwsza litera w nazwie pliku wskazuje poziom klasy:

- `a` – klasa 4
- `b` – klasa 8

Dalsze litery określają typ danych:

- `asa/bsa` – wyniki uczniów oraz wartości prawdopodobne (PV)
- `asp/bsp` – dane procesowe (np. czasy odpowiedzi)
- `ash` – dane z kwestionariusza rodzica
- `asg/bsg` – dane z kwestionariusza ucznia
- `acg/bcg` – dane z kwestionariusza szkoły
- `atg/btg` – dane z kwestionariusza nauczyciela

4.1.2 Struktura danych OECD

W przypadku badań OECD dane z różnych krajów są połączone w jeden zbiorczy plik (często bardzo duży), bez podziału na osobne pliki krajowe. Pliki są podzielone według cyklu badania oraz typu danych.

i Nazewnictwo plików w badaniach OECD, na przykładzie badania PISA

Przykład: Plik `CY08MSP_STU_QQQ.sav` zawiera dane z kwestionariuszy uczniów z wszystkich krajów z badania realizowanego w roku 2022.
Gdzie:

- `CY08` – cykl badania (tu: cykl 8, przeprowadzony w 2022 roku)
- `MSP` – Main Study (badanie główne)
- `STU_QQQ` – dane uczniów

Inne oznaczenia:

- `SCH_QQQ` – kwestionariusze szkół
- `TCH_QQQ` – kwestionariusze nauczycieli
- `STU_COG` – wyniki testów kognitywnych uczniów (czytanie, matematyka, nauki przyrodnicze itp.)
- `STU_FLT` – wyniki testów z edukacji finansowej
- `STU_ICT` – kwestionariusz dotyczący technologii informacyjno-komunikacyjnych
- `STU_WBQ` – kwestionariusz dobrostanu uczniów

5 Analiza danych PISA 2022

5.1 Przygotowanie danych

```
# Krok 1: wskazanie folderu z plikami SPSS
dir_pisa_2022 <- "C:/DATA/PISA 2022/Data_SPSS"

# Krok 2: wczytanie danych uczniowskich (STU_QQQ)
data_students <- read_sav(paste0(dir_pisa_2022, "/CY08MSP_STU_QQQ.sav"))

# Krok 3: Wybór krajów do analizy (Australia, Kanada, Peru, Polska) i
# filtrowanie zbioru danych
countries <- c("AUS", "CAN", "PER", "POL")
pisa_2022 <- filter(data_students, CNT %in% countries)

# Krok 4: zmiana wielkości liter w nazwach zmiennych na małe:
names(pisa_2022) <- tolower(names(pisa_2022))
```

5.2 Przykład 1 – średnia i odchylenie standardowe wyników umiejętności czytania według kraju i płci

```
Analysis_01 <- Rrepest(
  data = pisa_2022,
  svy  = "PISA",
  est  = est(c("mean", "std"), target = "pv@read"),
  by   = c("cnt", "st004d01t"),
  n.pvs = 10 #`Rrepest()` domyślnie używa 5 PV. W PISA 2022
             # należy ustawić `n.pvs = 10`
)
print(Analysis_01)
```

5.3 Przykład 2 – Tabela rozkładów częstości płci uczniów

```
Analysis_02 <- Rrepest(
  data = pisa_2022,
  svy  = "PISA",
  est  = est("freq", target = "st004d01t"),
  by   = "cnt",
  show_na = TRUE
)
print(Analysis_02, width = 200)
```

5.4 Przykład 3 – Regresja liniowa: analiza wpływu statusu społeczno–ekonomicznego rodziny na osiągnięcia ucznia w dziedzinie matematyki:

```
Analysis_03 <- Rrepest(
  data = pisa_2022,
  svy  = "PISA",
  by   = "cnt",
  est  = est("lm", target = "pv@math", regressor = "escs"),
  n.pvs = 10,
  show_na = TRUE
)
print(Analysis_03, width = 200)
```

Przykładowy kod poprawiający czytelność tabeli wynikowej

```
Table_03<- Analysis_03 %>%
  transmute(
    Country = cnt,
    Intercept_Estimate   = rowMeans(select(., starts_with("b.reg_pv") &
                           ends_with("intercept"))),
    Intercept_SE         = rowMeans(select(., starts_with("se.reg_pv") &
                           ends_with("intercept"))),
    Intercept_t          = Intercept_Estimate / Intercept_SE,

    ESCS_Estimate        = rowMeans(select(., starts_with("b.reg_pv") &
                           ends_with("escs"))),
    ESCS_SE              = rowMeans(select(., starts_with("se.reg_pv") &
```

```
                            ends_with("escs"))),
    ESCS_t                = ESCS_Estimate / ESCS_SE,

    R2_Estimate           = rowMeans(select(., starts_with("b.reg_pv") &
                            ends_with("rsqr"))),
    R2_SE                 = rowMeans(select(., starts_with("se.reg_pv") &
                            ends_with("rsqr"))),
    R2_t                  = R2_Estimate / R2_SE
  )

make_reg_table <- function(row) {
  data.frame(
    Estimate    = c(row$Intercept_Estimate,
                    row$ESCS_Estimate,
                    row$R2_Estimate),
    `Std. Error` = c(row$Intercept_SE,
                     row$ESCS_SE,
                     row$R2_SE),
    `t value`    = c(row$Intercept_t,
                     row$ESCS_t,
                     row$R2_t),
    row.names = c("(Intercept)", "ESCS", "R-squared")
  )
}

final_table_03 <- setNames(
  lapply(1:nrow(Table_03),
         function(i) make_reg_table(Table_03[i, ])),
  Table_03$Country
)

print(final_table_03)
```

5.5 Przykład 4 – Regresja logistyczna: wpływ statusu społeczno–ekonomicznego rodziny na osiągnięcia ucznia w dziedzinie matematyki (uzyskanie wyniku na poziomie co najmniej 626 punktów):

```
# Na potrzeby analizy z wykorzystaniem regresji logistycznej konieczne jest
# utworzenie nowej zmiennej przyjmującej wartości "1" lub "0", w zależności
# od tego czy dany uczeń osiągnął wynik równy lub wyższy od wartości 626 pkt
# (dla każdej imputacji PV tworzymy osobną zmienną).

for (i in 1:10) {
  varname <- paste0("pv", i, "math")
  newvar  <- paste0("y", i)
  pisa_2022[[newvar]] <- ifelse(pisa_2022[[varname]] >= 626, 1, 0)
}

Analysis_04 <- Rrepest(
  data = pisa_2022,
  svy  = "PISA",
  est  = est("log", target = "y@", regressor = "escs"),
  n.pvs = 10,
  by   = "cnt"
)
print(Analysis_04, width = 200)
```

Przykładowy kod poprawiający czytelność tabeli wynikowej

```
Table_04 <- Analysis_04 %>%
  transmute(
    Country   = cnt,
    Coef      = rowMeans(select(., starts_with("b.log")  &
                ends_with("intercept"))),
    Std_Error = rowMeans(select(., starts_with("se.log") &
                ends_with("intercept"))),
    t_value   = round(Coef / Std_Error, 2),
    ESCS      = rowMeans(select(., starts_with("b.log")  &
                ends_with("escs"))),
    ESCS_SE   = rowMeans(select(., starts_with("se.log") &
                ends_with("escs"))),
    ESCS_t    = round(ESCS / ESCS_SE, 2),
    OR        = round(exp(Coef), 2),
    CI95low   = round(exp(Coef - 1.96 * `Std_Error`), 2),
```

```
    CI95up    = round(exp(Coef + 1.96 * `Std_Error`), 2),
    OR_ESCS   = round(exp(ESCS), 2),
    CI95low_ESCS = round(exp(ESCS - 1.96 * `ESCS_SE`), 2),
    CI95up_ESCS  = round(exp(ESCS + 1.96 * `ESCS_SE`), 2)
  )

make_logreg_table <- function(row) {
  data.frame(
    Coef        = c(row$Coef, row$ESCS),
    `Std. Error`= c(row$Std_Error, row$ESCS_SE),
    `t value`   = c(row$t_value, row$ESCS_t),
    OR          = c(row$OR, row$OR_ESCS),
    CI95low     = c(row$CI95low, row$CI95low_ESCS),
    CI95up      = c(row$CI95up,  row$CI95up_ESCS),
    row.names = c("(Intercept)", "ESCS")
  )
}

final_table_04 <- setNames(lapply(1:nrow(Table_04),
                          function(i) make_logreg_table(Table_04[i, ])),
                          Table_04$Country)

print(final_table_04)
```

---

6 Analiza danych TIMSS 2023 (klasa 4)

6.1 Przygotowanie danych krok po kroku

Przygotowanie danych TIMSS do analizy za pomocą funkcji `Rrepest()` jest procesem wieloetapowym. Wynika to zarówno z charakterystyki struktury zbioru danych jak i metodologii badania TIMSS.

6.1.1 Połączenie danych

```
# Krok 1: wskazanie folderu z plikami SPSS
dir_timss_2023 <- "C:/DATA/TIMSS2023_IDB_SPSS_G4/2_Data Files/SPSS Data"

# Krok 2: utworzenie list plików dla wybranych krajów (Armenia, Australia,
# Bahrajn, Polska) oraz poszczególnych zbiorów danych:

# Pliki student achievement (ASA), zawierające wyniki testów umiejętności
timss_2023_asa_files <- list.files(dir_timss_2023,
  pattern = ".*ASA(POL|AUS|ARM|BHR).*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki student context (ASG), zawierające dane z ankiety uczniowskiej
timss_2023_asg_files <- list.files(dir_timss_2023,
  pattern = ".*ASG(POL|AUS|ARM|BHR).*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki school context (ACG), zawierające dane z ankiety dyrektorów szkół
timss_2023_acg_files <- list.files(dir_timss_2023,
  pattern = ".*ACG(POL|AUS|ARM|BHR).*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki home context (ASH), zawierające dane z ankiety rodziców
timss_2023_ash_files <- list.files(dir_timss_2023,
  pattern = ".*ASH(POL|AUS|ARM|BHR).*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Krok 3: Połączenie danych z ankiet z wybranych krajów z wykorzystaniem
# wcześniej utworzonych list plików
timss_2023_asa <- timss_2023_asa_files %>% map(read_sav) %>%
bind_rows()
```

```r
timss_2023_asg <- timss_2023_asg_files %>% map(read_sav) %>%
bind_rows()
timss_2023_acg <- timss_2023_acg_files %>% map(read_sav) %>%
bind_rows()
timss_2023_ash <- timss_2023_ash_files %>% map(read_sav) %>%
bind_rows()
```

> **!** Ważne
>
> W przypadku łączenia zbiorów z różnych kwestionariuszy w parametrze `"by"` funkcji `left_join()` podajemy wszystkie zmienne, które powtarzają się w łączonych zbiorach danych (w zależności od łączonych baz danych mogą to być takie zmienne jak identyfikator szkoły lub ucznia (odpowiednio np. "IDSCHOOL" i "IDSTUD") ale także wartości PV). W przeciwnym razie funkcja `left_join()` utworzy duplikaty kolumn zawierających te same zmienne, nadając nazwom tych zmiennych przyrostki (np. "zmienna01.x", "zmienna01.y"). Może to stanowić utrudnienie przy prowadzeniu analiz.

```r
# Krok 4: Połączenie zbiorów ze wszystkich kwestionariuszy
timss_2023 <- left_join(
  timss_2023_asa,
  select(timss_2023_acg, -any_of(names(timss_2023_asa)),
                                 CTY,
                                 IDSCHOOL),
  by = c("CTY", "IDSCHOOL")
) %>% left_join(
  select(timss_2023_asg, -any_of(names(timss_2023_asa)),
                                 CTY,
                                 IDSCHOOL,
                                 IDSTUD),
  by = c("CTY", "IDSCHOOL", "IDSTUD")
) %>% left_join(
  select(timss_2023_ash, -any_of(names(timss_2023_asa)),
                                 CTY,
                                 IDCNTRY,
                                 IDSCHOOL,
                                 IDSTUD),
  by = c("CTY", "IDCNTRY", "IDSCHOOL", "IDSTUD")
)
```

6.1.2 Utworzenie 250 wag replikacyjnych zgodnie z zastosowaną w badaniu TIMSS metodą Jackknife Repeated Replication (JRR)

```
# Krok 5: budowa 250 wag replikacyjnych metodą JRR
for (i in 1:125) {
  j <- 125 + i
  timss_2023 <- timss_2023 %>%
    mutate(!!paste0("jr", i) := case_when(
      JKZONE == i & JKREP == 1 ~ 2 * TOTWGT,
      JKZONE == i & JKREP == 0 ~ 0,
      TRUE ~ TOTWGT
    )) %>%
    mutate(!!paste0("jr", j) := case_when(
      JKZONE == i & JKREP == 0 ~ 2 * TOTWGT,
      JKZONE == i & JKREP == 1 ~ 0,
      TRUE ~ TOTWGT
    ))}
```

6.1.3 Zmiana wielkości liter w nazwach zmiennych

```
# Krok 6: zmiana wielkości liter w nazwach zmiennych na małe:
names(timss_2023) <- tolower(names(timss_2023))
```

> **!** Ważne
>
> **W analizach TIMSS/PIRLS**: należy użyć parametru custom weights **`cm.weights = c("totwgt", paste0("jr", 1:250))`** wskazując wagę główną **`totwgt`** oraz utworzone wcześniej wagi replikacyjne **`(jr1, ..., jr250)`**. Dla poprawnego przeliczenia błędów standardowych uwzględniających zastosowane wagi replikacyjne, konieczne jest także podanie wartości argumentu współczynnika wariancji: **`var.factor = 125`**. Wartość współczynnika wariancji zależna jest od zastosowanego schematu doboru próby. Aby zapewnić poprawność wykonywanych analiz, zapoznaj się z informacjami o metodzie tworzenia wag replikacyjnych zastosowanej w danym cyklu danego badania.

6.2 Przykład 1 – Średni wynik z matematyki według kraju i płci

```
Analysis_01 <- Rrepest(
  data = timss_2023,
  svy  = "TIMSS",
  cm.weights = c("totwgt", paste0("jr", 1:250)),
  est  = est(c("mean", "std"), target = "asmmat0@"),
  by   = c("cty", "itsex"),
  var.factor = 125
)
print(Analysis_01)
```

6.3 Przykład 2 – Średni wynik z matematyki według kraju i płci, z zastosowaniem parametru `over` i pokazaniem różnicy wyników pomiędzy płciami (chłopcy – dziewczęta) w poszczególnych krajach

```
Analysis_02 <- Rrepest(
  data = timss_2023,
  svy  = "TIMSS",
  cm.weights = c("totwgt", paste0("jr", 1:250)),
  est  = est("mean", target = "asmmat0@"),
  by   = "cty",
  over = "itsex",
  test = TRUE,
  var.factor = 125
)
print(Analysis_02, width = 200)
```

6.4 Przykład 3 – Tabela rozkładów częstości płci uczniów

```
Analysis_03 <- Rrepest(
  data = timss_2023,
  svy  = "TIMSS",
  cm.weights = c("totwgt", paste0("jr", 1:250)),
  est  = est("freq", "itsex"),
  by   = "cty",
  show_na = TRUE,
  var.factor = 125
)
print(Analysis_03)
```

6.5 Przykład 4 – Regresja liniowa: wpływ płci ucznia oraz braku wczesnych aktywności związanych z liczeniem na osiągnięcia uczniów w dziedzinie matematyki

```
Analysis_04 <- Rrepest(
  data = timss_2023,
  svy  = "TIMSS",
  cm.weights = c("totwgt", paste0("jr", 1:250)),
  by   = "cty",
  est  = est("lm", target = "asmmat0@", regressor = c("itsex", "asbh01k")),
  var.factor = 125
)
print(Analysis_04)
```

Przykładowy kod poprawiający czytelność tabeli wynikowej

```
Table_04 <- Analysis_04 %>%
  transmute(
    Country = cty,
    Intercept_Estimate = rowMeans(select(., starts_with("b.reg_asmmat") &
                         ends_with("intercept"))),
    Intercept_SE       = rowMeans(select(., starts_with("se.reg_asmmat") &
                         ends_with("intercept"))),
    Intercept_t        = Intercept_Estimate / Intercept_SE,
    ITSEX_Estimate     = rowMeans(select(., starts_with("b.reg_asmmat") &
                         ends_with("itsex"))),
```

```
    ITSEX_SE           = rowMeans(select(., starts_with("se.reg_asmmat") &
                         ends_with("itsex"))),
    ITSEX_t            = ITSEX_Estimate / ITSEX_SE,
    ASBH01K_Estimate   = rowMeans(select(., starts_with("b.reg_asmmat") &
                         ends_with("asbh01k"))),
    ASBH01K_SE         = rowMeans(select(., starts_with("se.reg_asmmat") &
                         ends_with("asbh01k"))),
    ASBH01K_t          = ASBH01K_Estimate / ASBH01K_SE,
    R2_Estimate        = rowMeans(select(., starts_with("b.reg_asmmat") &
                         ends_with("rsqr"))),
    R2_SE              = rowMeans(select(., starts_with("se.reg_asmmat") &
                          ends_with("rsqr"))),
    R2_t               = R2_Estimate / R2_SE
  )

make_reg_table <- function(row) {
  data.frame(
    Estimate     = c(row$Intercept_Estimate,
                     row$ITSEX_Estimate,
                     row$ASBH01K_Estimate,
                     row$R2_Estimate),
    `Std. Error` = c(row$Intercept_SE,
                     row$ITSEX_SE,
                     row$ASBH01K_SE,
                     row$R2_SE),
    `t value`    = c(row$Intercept_t,
                     row$ITSEX_t,
                     row$ASBH01K_t,
                     row$R2_t),
    row.names = c("(Intercept)", "ITSEX", "ASBH01K", "R-squared")
  )
}

final_table_04 <- setNames(lapply(1:nrow(Table_04),
                           function(i) make_reg_table(Table_04[i, ])),
                           Table_04$Country)

print(final_table_04)
```

6.6 Analizy na danych nauczycieli matematyki (TIMSS 2023)

Aby analizować dane dla nauczycieli musimy połączyć pliki z prefiksem ATG (teacher context) zawierające dane z ankiety nauczycielskiej z plikami z prefiksem AST (student-teacher linkage), gdyż w tym zbiorze znajdują się zmienne z wagami głównymi przypisanymi do nauczycieli ("TCHWGT" - wszyscy nauczyciele, "MATWGT" - nauczyciele matematyki, "SCIWGT" - nauczyciele nauk przyrodniczych)

6.6.1 Przygotowanie danych i budowa wag replikacyjnych dla nauczycieli zgodnie z zastosowaną w badaniu TIMSS 2023 metodą Jackknife Repeated Replication (JRR)

```
# Krok 1: Wczytanie danych z ankiet nauczycieli i uczniów dla wybranych krajów

# Pliki ATG - teacher context
# Pliki AST - student-teacher linkage

timss_2023_atg_files <- list.files(dir_timss_2023,
                                   pattern = ".*ATG(POL|AUS|ARM|BHR).*\\.sav$",
                                   ignore.case = TRUE,
                                   full.names = TRUE)

timss_2023_ast_files <- list.files(dir_timss_2023,
                                   pattern = ".*AST(POL|AUS|ARM|BHR).*\\.sav$",
                                   ignore.case = TRUE,
                                   full.names = TRUE)

timss_2023_atg <- timss_2023_atg_files %>% map(read_sav) %>% bind_rows()
timss_2023_ast <- timss_2023_ast_files %>% map(read_sav) %>% bind_rows()

# Krok 2: Połączenie baz danych
timss_2023_teacher <- left_join(
  timss_2023_atg,
  select(timss_2023_ast, -any_of(names(timss_2023_atg)),
                                 CTY,
                                 IDCNTRY,
                                 IDSCHOOL,
                                 IDTEACH,
                                 IDTEALIN),
  by = c("CTY", "IDCNTRY", "IDSCHOOL", "IDTEACH", "IDTEALIN")
)
```

```r
# Krok 3: Stworzenie 250 wag replikacyjnych z wykorzystaniem wagi nauczycieli
# matematyki (MATWGT)

for (i in 1:125) {
  j <- 125 + i
  timss_2023_teacher <- timss_2023_teacher %>%
    mutate(!!paste0("jr", i) := case_when(
      JKZONE == i & JKREP == 1 ~ 2 * MATWGT,
      JKZONE == i & JKREP == 0 ~ 0,
      TRUE ~ MATWGT
    )) %>%
    mutate(!!paste0("jr", j) := case_when(
      JKZONE == i & JKREP == 0 ~ 2 * MATWGT,
      JKZONE == i & JKREP == 1 ~ 0,
      TRUE ~ MATWGT
    ))
}

# Krok 4: Pozostawienie w zbiorze do dalszych analiz tylko nauczycieli
# matematyki (usuwamy wszystkie wiersze, gdzie zmienna "MATWGT" przyjmuje
# wartość NA)
timss_2023_teacher <- timss_2023_teacher %>% filter(!is.na(MATWGT))

# Krok 5: Zmiana wielkości liter w nazwach zmiennych
names(timss_2023_teacher) <- tolower(names(timss_2023_teacher))
```

! Ważne

W analizach dotyczących nauczycieli matematyki w badaniu TIMSS 2023 należy użyć parametru custom weights **`cm.weights = c("matwgt", paste0("jr", 1:250))`** wskazując wagę główną **`matwgt`** oraz utworzone wcześniej wagi replikacyjne **`(jr1, ..., jr250)`**. Dla poprawnego przeliczenia błędów standardowych uwzględniających zastosowane wagi, konieczne jest także podanie wartości argumentu współczynnika wariancji: **`var.factor = 125`**.

6.6.2 Przykład 5 – Odsetki nauczycieli matematyki według wieku

```
# rekodowanie wieku do zmiennej kategorycznej
timss_2023_teacher <- timss_2023_teacher %>%
  dplyr::mutate(
    atbg03 = factor(atbg03, levels = 1:6,
                    labels = c(
                    "Under 25",
                    "25-29",
                    "30-39",
                    "40-49",
                    "50-59",
                    "60 or more"))
  )

Analysis_05 <- Rrepest(
  data = timss_2023_teacher,
  svy  = "TIMSS",
  cm.weights = c("matwgt", paste0("jr", 1:250)),
  est  = est("freq", target = "atbg03"),
  by   = "cty",
  show_na = TRUE,
  var.factor = 125
)

print(Analysis_05)
```

6.6.3 Przykład 6 – Średnia długość stażu nauczycieli matematyki

```
Analysis_06 <- Rrepest(
  data = timss_2023_teacher,
  svy  = "TIMSS",
  cm.weights = c("matwgt", paste0("jr", 1:250)),
  est  = est("mean", target = "atbg01"),
  by   = "cty",
  var.factor = 125
)
print(Analysis_06)
```

7 Analiza danych ICCS 2022

> **!** Ważne
>
> Część krajów uczestniczących w badaniu ICCS 2022 po raz pierwszy przeprowadziła je w formie testów komputerowych. W krajach tych zrealizowano dodatkowy komponent badania w formie ankiet papierowych, tzw. „bridge study". Dane z próby „bridge" nie są uwzględniane w raportowaniu wyników ICCS 2022, dlatego w analizach pomijamy je, wybierając wyłącznie pliki z przyrostkiem „C4" w nazwie.

7.1 Przygotowanie danych

```
# Krok 1: wskazanie folderu z plikami SPSS
dir_iccs_2022 <- "C:/DATA/ICCS 2022/SPSS Data/ICCS_2022_G8"

# Krok 2: utworzenie list plików dla wybranych krajów (Chorwacja, Hiszpania,
# Polska, Rumunia, Szwecja) oraz poszczególnych zbiorów danych

# Uwaga: pomijamy pliki z edycji "bridge", wybierając pliki z przyrostkiem "C4"
# w nazwie

# Pliki Student Civic Knowledge (ISA), zawierające wyniki dotyczące kompetencji
# obywatelskich
iccs_isa_files <- list.files(dir_iccs_2022,
  pattern = ".*ISA(POL|HRV|ESP|SWE|ROU)C4.*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki International Student Questionnaire (ISG), zawierające dane z ankiety
# uczniowskiej
iccs_isg_files <- list.files(dir_iccs_2022,
  pattern = ".*ISG(POL|HRV|ESP|SWE|ROU)C4.*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki School Questionnaire (ICG), zawierające dane z ankiety
# dyrektorów szkół
iccs_icg_files <- list.files(dir_iccs_2022,
  pattern = ".*ICG(POL|HRV|ESP|SWE|ROU)C4.*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Pliki Teacher Questionnaire (ITG), zawierające dane z ankiety
```

```r
# nauczycielskiej
iccs_itg_files <- list.files(dir_iccs_2022,
  pattern = ".*ITG(POL|HRV|ESP|SWE|ROU)C4.*\\.sav$",
  ignore.case = TRUE, full.names = TRUE)

# Krok 3: Połączenie danych z ankiet z wybranych krajów z wykorzystaniem
# wcześniej utworzonych list plików
iccs_2022_isa <- iccs_isa_files %>% map(read_sav) %>% bind_rows()
iccs_2022_isg <- iccs_isg_files %>% map(read_sav) %>% bind_rows()
iccs_2022_icg <- iccs_icg_files %>% map(read_sav) %>% bind_rows()
iccs_2022_itg <- iccs_itg_files %>% map(read_sav) %>% bind_rows()
```

! Ważne

W przypadku badania ICCS 2022, przy łączeniu zbiorów danych zawierających wyniki dotyczące kompetencji obywatelskich uczniów (pliki z prefiksem ISA) oraz wyniki ankiety uczniowskiej (pliki z prefiksem ISG) uwzględniamy w argumencie `by` funkcji `left_join()` wartości PV, główną zmienną wagową oraz wagi replikacyjne uczniów. Jest to konieczne, ponieważ zmienne te występują w obydwu zbiorach.

```r
# Krok 4: Połączenie zbiorów danych dotyczących uczniów (wyniki dotyczące
# kompetencji obywatelskich oraz dane z ankiety uczniowskiej) oraz
# dyrektorów szkół

iccs_2022 <- left_join(
  iccs_2022_isa,
  iccs_2022_isg,
  by = c("COUNTRY", "IDSCHOOL", "IDSTUD", "std_GENDER",
         "TOTWGTS", paste0("PV", 1:5, "CIV"), paste0("SRWGTO", 1:9),
         paste0("SRWGT", 10:75))
) %>% left_join(
  iccs_2022_icg,
  by = c("COUNTRY", "IDSCHOOL")
)
```

! Ważne

W badaniu ICCS, możemy traktować zbiór nauczycielski jako niezależny od zbioru uczniowskiego i badać nauczycieli z wykorzystaniem wag nauczycielskich.

```r
# Nie łączymy danych nauczycieli z innymi zbiorami, ponieważ nie jest to
# potrzebne w naszych przykładowych analizach
iccs_2022_teacher <- iccs_2022_itg

# Krok 5: Zmiana wielkości liter w nazwach zmiennych
names(iccs_2022) <- tolower(names(iccs_2022))
names(iccs_2022_teacher) <- tolower(names(iccs_2022_teacher))
```

> **!** Ważne
>
> W wypadku badania ICCS, funkcja `Rrepest()` oczekuje, że zmienne wag replikacyjnych przyjmują nazwy bez zer prowadzących w przypadku replik o numeracji od 1 do 9 (np. "srwgt1,..., srwgt9" lub "trwgt1,..., trwgt9"). W udostępnionym przez IEA zbiorze danych ICCS 2022 nazwy zmiennych replikacyjnych zawierają zera prowadzące, przyjmując postać "srwgt01,..., srwgt09", "trwgt01,..., trwgt09". Aby funkcja zadziałała poprawnie, konieczna jest samodzielna zmiana nazw tych zmiennych.

```r
# Krok 6: usunięcie prowadzących zer w nazwach wag replikacyjnych
# w zbiorach uczniów i nauczycieli
colnames(iccs_2022) <- gsub("^srwgt0([1-9])$", "srwgt\\1", colnames(iccs_2022))
colnames(iccs_2022_teacher) <- gsub(
                               "^trwgt0([1-9])$",
                               "trwgt\\1",
                               colnames(iccs_2022_teacher))
```

7.2 Przykład 1 – Średnie wyniki uczniów w zakresie wiedzy i rozumienia kwestii obywatelskich w podziale na płeć

```r
Analysis_01 <- Rrepest(
  data = iccs_2022,
  svy  = "ICCS",
  est  = est(c("mean", "std"), target = "pv@civ"),
  by   = c("country", "std_gender")
)
print(Analysis_01)
```

7.3 Przykład 2 – Regresja liniowa: wpływ płci ucznia na wyniki w zakresie wiedzy i rozumienia kwestii obywatelskich

```
Analysis_02 <- Rrepest(
  data = iccs_2022,
  svy  = "ICCS",
  by   = "country",
  est  = est("lm", target = "pv@civ", regressor = c("std_gender"))
)
print(Analysis_02, width = 200)
```

Przykładowy kod poprawiający czytelność tabeli wynikowej

```
Table_02 <- Analysis_02 %>%
  transmute(
    country,
    Intercept_Estimate = rowMeans(select(., starts_with("b.reg_pv@civ") &
                         ends_with("intercept"))),
    Intercept_SE       = rowMeans(select(., starts_with("se.reg_pv@civ") &
                         ends_with("intercept"))),
    Intercept_t        = Intercept_Estimate / Intercept_SE,
    gender_Estimate    = rowMeans(select(., starts_with("b.reg_pv@civ") &
                         ends_with("std_gender"))),
    gender_SE          = rowMeans(select(., starts_with("se.reg_pv@civ") &
                         ends_with("std_gender"))),
    gender_t           = gender_Estimate / gender_SE,
    R2_Estimate        = rowMeans(select(., starts_with("b.reg_pv@civ") &
                         ends_with("rsqr"))),
    R2_SE              = rowMeans(select(., starts_with("se.reg_pv@civ") &
                         ends_with("rsqr"))),
    R2_t               = R2_Estimate / R2_SE
  )

make_reg_table <- function(row) {
  data.frame(
    Estimate     = c(row$Intercept_Estimate,
                     row$gender_Estimate,
                     row$R2_Estimate),
    `Std. Error` = c(row$Intercept_SE,
                     row$gender_SE,
                     row$R2_SE),
```

```
    `t value`    = c(row$Intercept_t,
                     row$gender_t,
                     row$R2_t),
    row.names = c("(Intercept)", "gender", "R-squared")
  )
}

Table_02 <- setNames(lapply(1:nrow(Table_02),
                     function(i) make_reg_table(Table_02[i, ])),
                     Table_02$country)
print(Table_02)
```

7.4 Przykład 3 – Partycypacja uczniów w życiu szkoły według dyrektorów szkół

7.4.1 Przykład 3A - Jak często uczniowie są zachęcani do włączania się w planowanie zajęć w klasie? Odsetek uczniów w szkołach, których dyrektorzy odpowiedzieli w określony sposób

```
Analysis_03a <- Rrepest(
  data = iccs_2022,
  svy  = "ICCS",
  est  = est("freq", target = "ic4g03d"),
  by   = "country"
)
print(Analysis_03a, width = 200)
```

7.4.2 Przykład 3B – Jak często uczniowie są zachęcani do włączania się w tworzenie szkolnych zasad i regulaminów? Odsetek uczniów w szkołach, których dyrektorzy odpowiedzieli w określony sposób

```
Analysis_03b <- Rrepest(
  data = iccs_2022,
  svy  = "ICCS",
  est  = est("freq", target = "ic4g03b"),
  by   = "country"
)
print(Analysis_03b, width = 200)
```

7.5 Przykład 4 - Odpowiedzi nauczycieli na pytanie: "Jak ważne dla bycia dobrym obywatelem w dorosłym życiu są następujące zachowania: Uczestniczenie w dyskusjach o polityce"

> **!** Ważne
>
> Aby funkcja `Rrepest()` automatycznie pobrała wagi przypisane do nauczycieli, należy ustawić wartość argumentu **'svy = "ICCS_T"'**. Jeżeli pozostawimy argument 'svy = "ICCS"', **konieczne** jest zastosowanie argumentu custom weights: **`cm.weights = c("totwgtt", paste0("trwgt", 1:75))`**.

```
Analysis_04 <- Rrepest(
  data = iccs_2022_teacher,
  svy  = "ICCS_T",
  est  = est("freq", target = "it4g17e"),
  by   = "country"
)
print(Analysis_04, width = 300)
```

---

8 Analiza danych TALIS 2018

W przypadku badania TALIS możemy prowadzić analizy na dwóch niezależnych poziomach:

- **TALISTCH** – nauczyciele (domyślne wagi: `TCHWGT` jako waga główna oraz **100** wag replikacyjnych `TRWGT1,…, TRWGT100`).
- **TALISSCH** – dyrektorzy/szkoły (domyślne wagi: `SCHWGT` jako waga główna oraz **100** wag replikacyjnych `SRWGT1,…, SRWGT100`).

Aby funkcja `Rrepest()` wczytała właściwe zmienne wagowe, konieczne jest poprawne ustawienie argumentu `svy`: **`svy = "TALISTCH"`** dla analiz wykonywanych na zbiorze nauczycielskim, lub **`svy = "TALISSCH"`** dla analiz wykonywanych na zbiorze szkolnym (dane z kwestionariusza dyrektorów szkół).

8.1 Przygotowanie danych

```
# Krok 1: wskazanie folderu z plikami SPSS
dir_talis_2018 <- "C:/DATA/TALIS 2018/SPSS Data/SPSS_2018_international"

# Krok 2: Wczytanie danych:

# Dane dotyczące nauczycieli - pliki z prefiksem BTG:
talis_2018_tch <- read_sav(file.path(dir_talis_2018, "BTGINTT3.sav"))

# Dane dotyczące dyrektorów szkół - pliki z prefiksem BCG:
talis_2018_sch <- read_sav(file.path(dir_talis_2018, "BCGINTT3.sav"))

# Krok 3: Wybór krajów do analizy (Czechy, Japonia, Kazachstan, Kolumbia)
# i filtrowanie zbiorów danych:

countries <- c("JPN", "CZE", "COL", "KAZ")

talis_2018_tch <- talis_2018_tch %>% filter(CNTRY %in% countries)
talis_2018_sch <- talis_2018_sch %>% filter(CNTRY %in% countries)

# Krok 4: zmiana wielkości liter w nazwach zmiennych na małe:
names(talis_2018_tch) <- tolower(names(talis_2018_tch))
names(talis_2018_sch) <- tolower(names(talis_2018_sch))
```

8.2 Analizy (TALISTCH - poziom nauczycieli)

8.2.1 Przykład 1 - Tabela rozkładów częstości płci nauczycieli

```
Analysis_01 <- Rrepest(
  data = talis_2018_tch,
  svy  = "TALISTCH",
  est  = est("freq", target = "tt3g01"),
  by   = "cntry"
)
print(Analysis_01)
```

8.2.2 Przykład 2 - Średnia liczba godzin poświęcanych przez nauczycieli tygodniowo na komunikację i współpracę z rodzicami

```
Analysis_02 <- Rrepest(
  data = talis_2018_tch,
  svy  = "TALISTCH",
  est  = est(c("mean","std"), target = "tt3g18h"),
  by   = "cntry"
)
print(Analysis_02)
```

8.2.3 Przykład 3 - Średnia liczba godzin poświęcanych przez nauczycieli tygodniowo na aktywności związane z rozwojem zawodowym

```
Analysis_03 <- Rrepest(
  data = talis_2018_tch,
  svy  = "TALISTCH",
  est  = est(c("mean","std"), target = "tt3g18g"),
  by   = "cntry"
)
print(Analysis_03)
```

8.2.4 Przykład 4 - Regresja liniowa: wpływ możliwości aktywnego uczestnictwa w decyzjach dotyczących szkoły na ogólny poziom satysfakcji nauczycieli z wykonywanej pracy

```
Analysis_04 <- Rrepest(
  data = talis_2018_tch,
  svy  = "TALISTCH",
  by   = "cntry",
  est  = est("lm", target = "t3jobsa", regressor = c("tt3g48a"))
)
print(Analysis_04, width = 200)
```

Przykładowy kod poprawiający czytelność tabeli wynikowej

```
Table_04 <- Analysis_04 %>%
  transmute(
    Country = cntry,
    Intercept_Estimate = rowMeans(select(., starts_with("b.reg_t3jobsa") &
                          ends_with("intercept"))),
    Intercept_SE       = rowMeans(select(., starts_with("se.reg_t3jobsa") &
                          ends_with("intercept"))),
    Intercept_t        = Intercept_Estimate / Intercept_SE,
    tt3g48a_Estimate   = rowMeans(select(., starts_with("b.reg_t3jobsa") &
                          ends_with("tt3g48a"))),
    tt3g48a_SE         = rowMeans(select(., starts_with("se.reg_t3jobsa") &
                          ends_with("tt3g48a"))),
    tt3g48a_t          = tt3g48a_Estimate / tt3g48a_SE,
    R2_Estimate        = rowMeans(select(., starts_with("b.reg_t3jobsa") &
                          ends_with("rsqr"))),
    R2_SE              = rowMeans(select(., starts_with("se.reg_t3jobsa") &
                          ends_with("rsqr"))),
    R2_t               = R2_Estimate / R2_SE
  )

make_reg_table <- function(row) {
  data.frame(
    Estimate     = c(row$Intercept_Estimate,
                     row$tt3g48a_Estimate,
                     row$R2_Estimate),
    `Std. Error` = c(row$Intercept_SE,
                     row$tt3g48a_SE,
                     row$R2_SE),
    `t value`    = c(row$Intercept_t,
                     row$tt3g48a_t,
                     row$R2_t),
    row.names    = c("(Intercept)", "tt3g48a", "R-squared")
  )
}

Final_04 <- setNames(lapply(1:nrow(Table_04),
                     function(i) make_reg_table(Table_04[i, ])),
                     Table_04$Country)

print(Final_04)
```

8.3 Analizy (TALISSCH - poziom szkół)

8.3.1 Przykład 5 - Średni staż pracy na stanowisku dyrektora szkoły w obecnym miejscu pracy (w latach)

```
Analysis_05 <- Rrepest(
  data = talis_2018_sch,
  svy  = "TALISSCH",
  est  = est(c("mean","std"), target = "tc3g04a"),
  by   = "cntry"
)
print(Analysis_05)
```

8.3.2 Przykład 6 - Odsetek dyrektorów szkół, którzy zadeklarowali, że współpracują z nauczycielami przy rozwiązywaniu problemów z dyscypliną w klasie

```
Analysis_06 <- Rrepest(
  data = talis_2018_sch,
  svy  = "TALISSCH",
  est  = est("freq", target = "tc3g22a"),
  by   = "cntry"
)
print(Analysis_06, width = 200)
```

---

9 Podsumowanie

Pakiet **Rrepest** może być traktowany jako dogodna alternatywa dla programu **IEA IDB Analyzer** w zakresie wykonywania analiz danych z międzynarodowych badań edukacyjnych, szczególnie dla osób pracujących głównie w środowisku R. Rrepest umożliwia automatyzację wielu powtarzalnych procedur, zapewniając obsługę wielu badań edukacyjnych oraz zgodność przeprowadzonych analiz z metodologią badań OECD i IEA. Jednocześnie należy zaznaczyć, że pakiet ten nie obejmuje pełnego zakresu analiz dostępnych np. w **IEA IDB Analyzer** lub w **pakiecie intsvy**.

Zachęcamy do wykorzystywania pakietu we własnych projektach badawczych, testowania jego funkcjonalności w analizie danych z międzynarodowych badań edukacyjnych oraz do zgłaszania uwag i propozycji usprawnień bezpośrednio do autorów pakietu poprzez **repozytorium GitLab**.

10 Bibliografia

- Avvisati, F., & Keslair, F. (2014). *REPEST: Stata module to run estimations with weighted replicate samples and plausible values* (Statistical Software Components S457918). Boston College Department of Economics. (Wersja poprawiona: 11 grudnia 2024 r.).
- Ilizaliturri, R., Avvisati, F., & Keslair, F. (2025). *Rrepest: An analyzer of international large-scale assessments in education* (Version 1.5.4) [R package]. CRAN. https://CRAN.R-project.org/package=Rrepest