IBE losy absolwentów szkół zawodowych

Tomasz Żółtak

Monitorowanie Losów Edukacyjno-Zawodowych Absolwetnów i Młodych Dorosłych - oprogramowanie

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
IBE losy absolwentów szkół zawodowych
Unia Europejska Europejski Fundusz Społeczny

Warszawa 2019

Autor:
dr Tomasz Żółtak

Wzór cytowania:
Żółtak Tomasz. (2019). *Monitorowanie Losów Edukacyjno-Zawodowych Absolwentów i Młodych Dorosłych - oprogramowanie*. Warszawa: Instytut Badań Edukacyjnych.

Wydawca:
Instytut Badań Edukacyjnych
ul. Górczewska 8
01-180 Warszawa
tel. (22) 241 71 00; www.ibe.edu.pl

Egzemplarz bezpłatny

Streszczenie

Raport skupia się na opisie i wyjaśnieniu założeń funkcjonalnych oraz typowych zastosowań rozwiązań informatycznych opracowanych w ramach projektu Monitorowanie Losów Edukacyjno-Zawodowych Absolwentów i Młodych Dorosłych (MLEZAiMD). Służą one do obliczania wskaźników charakteryzujących sytację edukacyjno-zawodową absolwentów oraz automatycznemu generowaniu raportów zawierających zestawienia wartości tych wskaźników. W raporcie opisane zostały również przeprowadzone testy oprogramowania, potwierdzające realizowanie przez niego założonych funkcjonalności.

Spis Treści

1. Wprowadzenie

W ramach prac nad opracowaniem, przetestowaniem i przygotowaniem do wdrożenia systemowych narzędzi pozwalających na systematyczne śledzenie i monitorowanie losów zawodowych absolwentów szkół zawodowych w latach 2016-2019 Instytut Badań Edukacyjnych przeprowadził projekt „Monitorowanie Losów Edukacyjno-Zawodowych Absolwentów i Młodych Dorosłych" (MLEZAiMD, sygnatura: POWR.02.15.00-IP.02-00-004/16). Jednym z jego celów było opracowanie programów umożliwiających przekształcanie zbiorów danych zawierających informacje o losach absolwentów szkół zawodowych, obliczanie na ich podstawie wskaźników i generowanie raportów szkół zawierających zestawienia wartości tych wskaźników. Programy te przygotowane zostały w formie pakietów środowiska R. Pakiet *MLASZdane* odpowiada za przygotowanie zbiorów danych z pilotażowej i z pierwszej rundy monitoringu losów absolwentów oraz obliczenie na ich podstawie wartości wskaźników charakteryzujących sytuację absolwentów. Pakiet *MLASZraporty* odpowiada za przygotowanie raportów szkół na podstawie zbiorów danych zawierających wartości wskaźników.

W kolejnych częściach raportu przedstawione zostały: architektura opracowanych programów i przygotowywanych zbiorów danych, typowe sposoby użycia programów oraz zakres testów przeprowadzonych w celu zapewnienia poprawności ich działania. Trzeba przy tym zaznaczyć, że wbrew założeniom przyjętym na początku projektu, w konsekwencji problemów prawnych, ani w ramach pilotażowej, ani w ramach pierwszej rundy monitoringu losów absolwentów szkół zawodowych nie udało się pozyskać danych absolwentów z rejestrów państwowych, ani baz danych administracyjnych. W związku z tym funkcjonalności pakietu *MLASZdane* w obecnej wersji odnoszą się do przekształcanie danych pozyskanych w wyniku realizacji badań sondażowych, które były elementem pilotażowej i pierwszej rundy monitoringu losów absolwentów szkół zawodowych (sposób realizacji tych badań opisany został szczegółowo w odrębnym *Raporcie metodologicznym* z pierwszej rundy monitoringu). Jednocześnie przygotowane programy umożliwiają obliczenie również takich wskaźników charakteryzujących sytuację absolwentów, które nie mogłyby być wykorzystane, gdyby monitoring prowadzony był wyłącznie z wykorzystaniem danych administracyjnych (tj. opisujących poglądy absolwentów).

2. Architektura przyjętych rozwiązań informatycznych

2.1. Założenia funkcjonalne

Przyjęte założenia funkcjonalne opracowanych programów wynikają z kilku przesłanek. Pierwszą z nich jest dążenie do budowania rozwiązań informatycznych służących monitorowaniu losów absolwentów szkół zawodowych w sposób kompatybilny z rozwiązaniami już wdrożonymi w ramach systemu monitorowania losów absolwentów szkół wyższych (system ELA). Jeżeli byłoby to możliwe, wskazane byłoby wręcz wykorzystanie programów, które zostały w ramach tego systemu opracowane (są to

programy na otwartej licencji, więc działania takie nie napotykają przeszkód prawnych). Umożliwiłoby to ograniczenie zakresu koniecznych prac i zmniejszenie ryzyka napotkania problemów w implementacji. Ujednolicenie metody obliczania przynajmniej części wskaźników pomiędzy systemami monitorowania losów absolwentów szkół zawodowych i szkół wyższych mogłoby też sprzyjać szerszej percepcji przygotowywanych wskaźników i czyniło je bardziej użytecznymi do celów porównawczych. Z drugiej strony, w ramach projektowanych rozwiązań należało uwzględnić specyfikę edukacji na poziomie średnim, w szczególności inną organizację systemu placówek edukacyjnych, większe rozdrobnienie, jak też nieco inne oczekiwania odbiorców względem zakresu i formy prezentowanych informacji. W tym ostatnim aspekcie zidentyfikowano zwłaszcza potrzebę możliwie szerokiego wykorzystania wykresów do prezentacji wartości wskaźników oraz prezentowania wartości wskaźnika w danej szkole w połączeniu z wartością dla odpowiednio dobranego punktu odniesienia (np. wartości w grupie podobnych szkół). Ostatnia istotna przesłanka, która wpłynęła na modyfikację przyjętych założeń funkcjonalnych zaistniała już w trakcie realizacji projektu – była nią niemożliwość pozyskania danych administracyjnych opisujących sytuację absolwentów. W związku z tym konieczne było skupienie się w ramach realizowanych prac nad wykorzystaniem danych zebranych w wyniku realizacji badań sondażowych, które według pierwotnych założeń miały posłużyć przede wszystkim do badania trafności wskaźników obliczanych na podstawie danych administracyjnych. W konsekwencji przyjęte zostały następujące założenia funkcjonalne:

- Programy powinny zostać przygotowane w formie pakietów środowiska R, w celu zapewnienia kompatybilności technologicznej z oprogramowaniem wykorzystywanym w systemie monitorowania absolwentów szkół wyższych. Powinny też korzystać w duże mierze z podobnych bibliotek/technologii w ramach tych dostępnych dla środowiska R (np. pakietu *dplyr* do przekształcania danych, czy pakietów *knitr* i *rmarkdown* w połączeniu z programem *pandoc* do generowania raportów).

- Jeden z opracowanych programów – *MLASZdane* – powinien odpowiadać za przygotowanie danych i obliczanie wartości wskaźników (w tym również na poziomie zagregowanym, w szczególności na poziomie szkół), drugi zaś – *MLASZraporty* – jedynie za przygotowywanie raportów na podstawie zestawień wskaźników na poziomie zagregowanym.

- Funkcjonalności pakietu *MLASZdane* powinny obejmować:

    - Przetwarzanie zbiorów danych opisujących wyniki sondażu do formy bardziej użytecznej analitycznie i jednocześnie bardziej zbliżonej do tej, w jakiej mogłyby zostać otrzymane dane administracyjne (przede wszystkim z ZUS).

    - Obliczanie wskaźników charakteryzujących sytuację absolwenta na podstawie danych sondażowych, z uwzględnieniem również tych informacji, których pozyskanie z danych administracyjnych byłoby niemożliwe (w szczególności oceny satysfakcji z wykonywanej pracy oraz jej zgodności z kierunkiem kształcenia).

- Pobieranie danych z Banku Danych Lokalnych GUS za pośrednictwem HTTP REST API udostępnianego przez gestora tej bazy.

- Pakiet *MLASZraporty:*

  - Pakiet powinien być jak najprostszy w użyciu. W szczególności powinien zawierać gotowe szablony raportów (w wersji przygotowanej w ramach projektu MLEZAiMD co najmniej jeden szablon raportu, pozwalający generować raporty szkół na podstawie danych z pierwszej rundy monitoringu losów absolwentów).

  - Przygotowane szablony raportów powinny w dużym stopniu wykorzystywać graficzne formy prezentacji wskaźników.

  - Aby ułatwić interpretację raportów, wartości wskaźników opisujących daną grupę badanych co do zasady powinny być prezentowane w zestawieniu z wartościami tych samych wskaźników w *grupie porównawczej*. Pakiet powinien wspierać wykorzystanie szablonów raportów wykorzystujących informacje o wartościach wskaźników w *grupach porównawczych*.

  - Pakiet powinien zapewniać automatyczne generowanie raportów na podstawie szablonów zarówno do formatu PDF, jak i do formatu HTML.

W związku z przyjęciem takiej architektury zdecydowano się na specyficzną strukturę zbiorów danych ze wskaźnikami zagregowanymi *przekazywanych* pomiędzy pakietami *MLASZdane* i *MLASZwykresy*. W odróżnieniu od wskaźników obliczanych na poziomie indywidualnym, wskaźniki te nie są typowymi zmiennymi liczbowymi lub tekstowymi, lecz są tworzone w formie obiektów o bardziej złożonej strukturze, tzw. kolumn-list. Dla każdego wiersza element takiej kolumny-listy sam jest listą (o takiej samej strukturze dla każdego wiersza). Umożliwia to zawarcie w zbiorze danych zagregowanych znacznej części informacji o znaczeniu zapisywanych w nim wartości (wykorzystywane jest tu nadawanie nazw elementom listy) i logicznych powiązaniach pomiędzy różnymi wartościami opisującymi poszczególne aspekty tej samej charakterystyki grupy (powiązanie takie definiuje uwzględnienie poszczególnych wartości jako elementów tego samego wskaźnika). Umożliwia to uzyskanie względnie dobrej orientacji, co do znaczenia i sposobu rozsądnej interpretacji wskaźników zawartych w zbiorze zagregowanym na podstawie samej tylko struktury tego zbioru. Dzięki temu tworzenie szablonów raportów wykorzystujących takie zbiory staje się łatwiejsze, czytelniejsza jest też struktura przygotowanych szablonów (choć dzieje się to niekiedy kosztem zwięzłości kodu, jako że opisowe nazwy elementów wskaźników bywają długie). Konsekwencją takiej złożonej struktury zbiorów danych ze wskaźnikami na poziomie zagregowanym konieczne jest zapisywanie ich w formie plików .RData (plików binarnych przeznaczonych do zapisu obiektów środowiska R).

2.2. Procedura obliczania wskaźników

Przygotowanie wskaźników charakteryzujących losy edukacyjno-zawodowe absolwentów poszczególnych szkół jest złożonym procesem, w którym można wyodrębnić następujące etapy:

1. Obróbka i integracja danych dotyczących indywidualnych absolwentów.

2. Obliczenie wartości wskaźników charakteryzujących sytuację edukacyjno-zawodową indywidualnych absolwentów.

3. Obliczenie wartości wskaźników charakteryzujących sytuację edukacyjno-zawodową grup absolwentów.

Pierwszy z ww. etapów docelowo będzie obejmować pracę ze zbiorami danych wyeksportowanymi z rejestrów i baz danych administracyjnych. Typowo będą to zbiory danych, w których jednego absolwenta może opisywać wiele rekordów (*osobo-epizodów* lub *osobo-miesięcy* – por. rozdział 2.3). W ramach pierwszej rundy monitoringu zrealizowane tu działania polegały na kontroli jakości i obróbce zbiorów z wynikami przeprowadzonych badań sondażowych, w celu nadania im właśnie takiej formy, umożliwiającej łatwe obliczenie wskaźników charakteryzujący poszczególnych absolwentów. Opracowane procedury będą mogły być, po niewielkich adaptacjach, zastosowane również w kolejnych rundach monitoringu, w których informacje o losach absolwentów będą zbierane przy pomocy metod sondażowych. Z drugiej strony, uzyskanie dostępu dodanych z rejestrów i baz danych administracyjnych będzie wymagało opracowania nowych rozwiązań zapewniających realizację tego etapu prac, zapewne bardziej bezpośrednio wzorowanych na tych, które wykorzystywane są w systemie monitorowania losów absolwentów szkół wyższych.

Drugi etap obliczania wskaźników obejmuje przekształcenia, w wyniku których otrzymywany jest zbiór, w którym każdemu z absolwentów odpowiada tylko jeden rekord, a poszczególne zmienne-wskaźniki opisują istotne aspekty jego sytuacji edukacyjno-zawodowej w konkretnym momencie (zestaw wskaźników obliczonych na podstawie badań zrealizowanych w ramach pierwszej rundy monitoringu opisany został szczegółowo w dokumentacji pakietu *MLASZdane* oraz w raporcie metodologicznym z pierwszej rundy monitoringu; można przyjąć, że w kolejnych rundach monitoringu nie powinien on ulegać znacznym zmianom).

Ostatni etap przygotowywania wskaźników wykorzystywanych w raportach dla szkół to agregacja wskaźników charakteryzujących sytuację poszczególnych absolwentów, tak aby w syntetyczny sposób charakteryzowały one sytuację kończących daną szkołę. Aby ułatwić interpretację wartości wskaźników prezentowanych w raportach, zdecydowano się przy tym obliczać je nie tylko na poziomie szkół, ale również szerszych grup – w pierwszej rundzie monitoringu jest to typ szkoły zawodowej. Mają one za zadanie służyć jako *grupy porównawcze*, pomagając ustalić punkt odniesienia do oceny wartości wskaźnika w poszczególnych szkołach. W tym celu wartość każdego wskaźnika w odpowiedniej *grupie porównawczej* jest prezentowana w raportach szkół obok wartości wskaźnika w danej placówce.

2.3. Przetwarzanie zbiorów danych z wynikami badań sondażowych

Informacje o historii edukacyjnej i zawodowej absolwentów po ukończeniu przez nich szkoły zbierane były w badaniu w specyficzny sposób: respondentów proszono o wymienianie kolejnych szkół (uczelnie), do których starali się dostać i w których

kontynuowali naukę, kolejnych miejsc (i stanowisk) pracy itp. Odnośnie do każdego z takich wymienionych *epizodów* historii edukacyjno-zawodowej zadawane były respondentowi pytania mające na celu zebranie bardziej szczegółowych informacji nt. tego, gdzie, jak długo, w jakiej formie i dlaczego się on uczył, pracował lub poszukiwał pracy. W zbiorze danych z wynikami badania (w którym jednego respondenta opisuje jeden wiersz – jest to tzw. *postać szeroka*) zmienne, w których zakodowano odpowiedzi na te pytania, występują w formie, która utrudnia ich analizę: każdy kolejny *epizod* danego typu opisują inne zmienne, mimo że zawierają one analogiczne informacje. Niełatwe jest też śledzenie następstwa czasowego pomiędzy *epizodami* różnych typów.

Rozwiązanie problemu analizy danych odnoszących się do *epizodów* stanowi przekształcenie ich w ten sposób, aby pojedynczy wiersz opisującego je zbioru danych odnosił się do konkretnego *epizodu*. Co za tym idzie, historia edukacyjno-zawodowa jednego respondenta może być w takim zbiorze opisywana przez wiele wierszy (jest to tzw. *postać długa* zbioru danych). Analizowanie danych przechowywanych w takiej formie ma kilka istotnych zalet:

- Pozwala łatwo porównywać ze sobą te same cechy kolejnych *epizodów* tego samego typu, które następują w historii edukacyjno-zawodowej tego samego respondenta.

- Pozwala łatwo zestawiać ze sobą podstawowy zestaw cech odnoszących się do każdego typu *epizodów* (lub przynajmniej kilku różnych typów): typ, czas rozpoczęcia i zakończenia, czy zakończył się on *sukcesem* (w przypadku edukacji).

- Pozwala łatwo śledzić następstwo zdarzeń w historii edukacyjno-zawodowej respondenta.

- Pozwala w łatwy sposób, przy pomocy metod agregacji danych, tworzyć nowe zmienne charakteryzujące w syntetyczny sposób interesujące analitycznie cechy historii edukacyjno-zawodowej poszczególnych respondentów.

Aby umożliwić łatwe prowadzenie analiz wykorzystujących tego typu porównania (przekształcenia), pakiet *MLASZdane* umożliwia utworzenie zbioru danych opisującego wyniki badania w formie, w której jednostkami obserwacji są poszczególne *epizody* (p. rozdział 3.2.2). Na podstawie tak przekształconego zbioru danych obliczane są później wskaźniki indywidualne opisujące sytuację poszczególnych absolwentów.

Kolejną trudność specyficzną dla danych sondażowych stanowi występowanie braków danych wynikających z odmowy odpowiedzi lub trudności z jej udzieleniem. Ich występowanie jest szczególnie kłopotliwe w odniesieniu do dat uzyskania, zmiany lub utraty pracy, podjęcia lub zakończenia nauki itp. Aby częściowo zaradzić temu problemowi, w pakiecie *MLASZdane* zaimplementowane zostały procedury imputacji braków danych odnoszących się do dat rozpoczęcia i zakończenia poszczególnych *epizodów*. Zostały one szczegółowo opisane w dokumentacji pakietu oraz raporcie metodologicznym z pierwszej rundy monitoringu. Sposób wykorzystania pakietu do przeprowadzenia takiej imputacji opisany został z kolei w dalszej części tego raportu, w rozdziale 3.2.2. Należy przeprowadzić ją przed przystąpieniem do obliczania na

podstawie zbioru *epizodów* wskaźników indywidualnych opisujących sytuację absolwentów.

3. Typowe sposoby użycia

3.1. Instalacja oprogramowania

Pakiety *MLASZdane* i *MLASZraporty* napisane są w języku R i wymagają do działania środowiska tego języka. Pliki instalacyjne środowiska R dostępne są pod adresem:

- dla Windows *https://cran.r-project.org/bin/windows/base/*
- dla Mac OS *https://cran.r-project.org/bin/macosx/*
- dla Linuxa *https://cran.r-project.org/bin/linux/* lub w paczkach poszczególnych dystrybucji.

Aby móc generować raporty z użyciem *MLASZraporty*, niezbędne jest zainstalowanie dodatkowo wymienionych poniżej programów. Wszystkie potrzebne one są darmowe oraz dostępne zarówno dla Windows, MacOs, jak i Linuxa.

- ***RStudio*** – dostępny pod adresem http://www.rstudio.com/products/rstudio/download/ (dla wszystkich platform).
  - *RStudio* to zintegrowane środowisko programistyczne (ang. IDE) dedykowane programowi R. Posługiwanie się nim jest zalecane (choć nie jest konieczne) również przy korzystaniu z pakietu *MLASZdane*.
- ***Pandoc*** – jest instalowany razem z programem *RStudio*.
- Dowolna ***dystrybucja LaTeX-a***, np.:
  - pod Windows *MiKTeX*: *http://miktex.org/download*
  - pod Mac OS *MacTeX*; *https://tug.org/mactex/*
  - pod Linuxem *TeX Live*: dostępny w paczkach dystrybucji.

Po uruchomienie środowiska R konieczne jest jeszcze zainstalowanie w nim pakietów *MLASZdane* i *MLASZraporty*. Instalację najprościej przeprowadzić wykorzystując pakiet *devtools*. W tym celu w konsoli programu *RStudio* należy wywołać komendy:

```
# potrzebne tylko, gdy pakiet devtools
# nie jest jeszcze zainstalowany
install.packages('devtools')
# instalacja MLASZdane
devtools::install_github('tzoltak/MLASZdane')
# instalacja MLASZraporty
devtools::install_github('tzoltak/MLASZraporty')
```

Dokładnie w ten sam sposób można przeprowadzić aktualizację pakietów do najnowszej wersji.

3.2. Wykorzystane pakietu *MLASZdane* do przygotowania zbiorów z badań sondażowych do analizy

3.2.1. Przygotowanie zbiorów z pierwszej rundy monitoringu

Do przetwarzania wyników badań sondażowych przeprowadzonych w projekcie MLEZAiMD w ramach pierwszej rundy monitoringu (zbiór danych: *MLEZAiMD_I_runda_CAPI_absolwent_n7713_20180924_z_wagami_z_kodowaniem.sav*) do postaci, na podstawie której możliwe jest łatwe obliczenie wskaźników charakteryzujących sytuację edukacyjno-zawodową absolwentów, służą następujące funkcje:

`wczytaj_wyniki_1rm()`– wczytuje zbiór danych z wynikami badania CAPI absolwentów (zapisany w formacie .sav programu SPSS) i tworzy na jego podstawie:

- zbiór wymienionych przez respondentów epizodów nauki/pracy/bezrobocia (o które w czasie wywiadu pytano w pętlach, prosząc o wymienienie po kolei wszystkich epizodów danego rodzaju) w formie długiej (tj. wiele wierszy opisuje jednego respondenta – po jednym wierszu na każdy wymieniony przez niego epizod),
- zbiór członków gospodarstw domowych w formie długiej (tj. wiele wierszy może opisywać gospodarstwo domowe tego samego respondenta – po jednym wierszu na każdego członka gospodarstwa domowego),
- zbiór z pozostałymi zmiennymi;

`imputuj_miesiac_pk_1rm()`– imputuje wartości zmiennych opisujących czas rozpoczęcie i zakończenia się poszczególnych epizodów nauki/pracy/bezrobocia, jeśli respondent, odpowiadając na pytania, nie określił ich w sposób precyzyjny;

`przygotuj_zbior_osobo_miesiecy_1rm()`– tworzy zbiór, w którym jednostką obserwacji jest status nauki/zatrudnienia respondenta w danym miesiącu od ukończenia szkoły, użyteczny do analizy i ilustrowania zmiany statusu nauki/zatrudnienia absolwentów w czasie.

Wszystkie ww. funkcje przygotowują zbiory danych w taki sposób, że możliwe jest łatwe zapisanie ich do plików w formacie .sav, korzystając z funkcji pakietu *haven*.

Schemat wykorzystania ww. funkcji wygląda następująco:

```
library(MLASZdane)
# wczytanie i przetworzenie zbioru z wynikami sondażu
dane1RM =
wczytaj_wyniki_1rm("MLEZAiMD_I_runda_CAPI_absolwent_n7713_20180924
_z_wagami_z_kodowaniem.sav")
## wynikiem działania jest lista zbiorów
str(dane1RM)
```

```
head(dane1RM$epizody)
head(dane1RM$gospDom)

# imputowanie czasów rozpoczęcia i zakończenia epizodów
dane1RM = imputuj_miesiac_pk_1rm(dane1RM)

# zapis utworzonych zbiorów do plików w formacie .sav
library(haven)
write_sav(dane1RM$epizody, "MLEZAiMD_runda1_epizody.sav")
write_sav(dane1RM$gospDom, "MLEZAiMD_runda1_gosp_dom.sav")
write_sav(dane1RM$dane, "MLEZAiMD_runda1_bez_petli.sav")

# tworzenie i zapis pliku osobo-miesięcy
osMies1RM = przygotuj_zbior_osobo_miesiecy_pilrm(dane1RM)
head(osMies1RM)
write_sav(osMies1RM, "MLEZAiMD_runda1_osobo-miesiace.sav")
```

3.2.2. Przygotowanie zbiorów z pilotażowej rundy monitoringu

Do przetwarzania wyników badań sondażowych przeprowadzonych w projekcie MLEZAiMD w ramach pilotażowej rundy monitoringu (zbiór danych: *MLEZAMiD_absolwent_n2959_20171013.sav*) służą następujące funkcje:

- `wczytaj_wyniki_pilrm()` – wczytuje zbiór danych z wynikami badania CAPI absolwentów (zapisany w formacie .sav programu SPSS) i tworzy na jego podstawie:
  - zbiór wymienionych przez respondentów epizodów nauki/pracy/bezrobocia (o które w czasie wywiadu pytano w pętlach, prosząc o wymienienie po kolei wszystkich epizodów danego rodzaju) w formie długiej (tj. wiele wierszy opisuje jednego respondenta – po jednym wierszu na każdy wymieniony przez niego epizod),
  - zbiór członków gospodarstw domowych w formie długiej (tj. wiele wierszy może opisywać gospodarstwo domowe tego samego respondenta - po jednym wierszu na każdego członka gospodarstwa domowego),
  - zbiór czasów odpowiedzi na pytania,
  - zbiór z pozostałymi zmiennymi;
- `imputuj_miesiac_pk_pilrm()` – imputuje wartości zmiennych opisujących czas rozpoczęcie i zakończenia się poszczególnych epizodów nauki/pracy/bezrobocia, jeśli respondent, odpowiadając na pytania, nie określił ich w sposób precyzyjny;
- `przygotuj_zbior_osobo_miesiecy_pilrm()` – tworzy zbiór, w którym jednostką obserwacji jest status nauki/zatrudnienia respondenta w danym miesiącu od ukończenia szkoły, użyteczny do analizy i ilustrowania zmiany statusu nauki/zatrudnienia absolwentów w czasie.

Wszystkie ww. funkcje przygotowują zbiory danych w taki sposób, że możliwe jest łatwe zapisanie ich do plików w formacie .sav, korzystając z funkcji pakietu *haven*.

Schemat wykorzystania ww. funkcji wygląda następująco:

```
library(MLASZdane)

# wczytanie i przetworzenie zbioru z wynikami sondażu
danePilRM =
wczytaj_wyniki_pilrm("MLEZAMiD_absolwent_n2959_20171013.sav")
## wynikiem działania jest lista zbiorów
str(danePilRM)
head(danePilRM$epizody)
head(danePilRM$gospDom)

# imputowanie czasów rozpoczęcia i zakończenia epizodów
danePilRM = imputuj_miesiac_pk_pilrm(danePilRM)

# zapis utworzonych zbiorów do plików w formacie .sav
library(haven)
write_sav(danePilRM$epizody, "MLEZAiMD_runda_pilot_epizody.sav")
write_sav(danePilRM$gospDom, "MLEZAiMD_runda_pilot_gosp_dom.sav")
write_sav(danePilRM$dane, "MLEZAiMD_runda_pilot_bez_petli.sav")

# tworzenie i zapis pliku osobo-miesięcy
osMiesPilRM = przygotuj_zbior_osobo_miesiecy_pilrm(danePilRM)
head(osMiesPilRM)
write_sav(osMiesPilRM, "MLEZAiMD_runda_pilot_osobo-miesiace.sav")
```

3.3. Wykorzystanie pakietu *MLASZdane* do pobrania wskaźników z API BDL GUS

Pakiet *MLASZdane* umożliwia też pobieranie wskaźników opisujących sytuację na rynku pracy w jednostkach samorządu terytorialnego (JST) z API Banku Danych Lokalnych (BDL) GUS. Wskaźniki te są konieczne jako informacja uzupełniająca przy tworzeniu wskaźników charakteryzujących grupy absolwentów w odniesieniu do poziomu bezrobocia i wysokości wynagrodzeń (mogłyby też zostać wykorzystane do tworzenia tzw. *wskaźników względnych*, zgodnie z metodą wykorzystywaną w systemie monitorowania losów absolwentów szkół wyższych ELA, choć w odniesieniu do uczniów szkół zawodowych podejście takie nie zostało wykorzystane). Pobieranie wartości wskaźników z BDL GUS przebiega w dwóch krokach:

1. Wyszukanie wskaźników, które chce się pobrać przy pomocy funkcji `znajdz_wskazniki_bdl()` lub `wskaznik_bdl()`.

2. Pobranie wartości wyszukanych wcześniej wskaźników przy pomocy funkcji `pobierz_dane_bdl()`.

3.3.1. Wyszukiwanie wskaźników

Wyszukiwanie wskaźników BDL przy użyciu API nie jest łatwe ze względu na przyjęty przez GUS nie do końca spójny (a przynajmniej niezbyt wygodny) schemat nazywania wskaźników. Wskaźniki można bowiem podzielić na dwie grupy: te, które posiadają

unikalne i informatywne nazwy oraz te, których nazwy są nieinformatywne i nie są unikalne (w szczególności wskaźniki o nazwie „ogółem”). Te pierwsze można łatwo wyszukiwać po nazwach (korzystając z odpowiedniej funkcji API), te drugie trzeba znajdować, przeszukując krok po kroku w głąb drzewiastą strukturę grup i podgrup wskaźników. Ponieważ to drugie podejście jest dosyć skomplikowane i słabo poddające się automatyzacji, w ramach pakietu zdecydowano się przyjąć następujące podejście:

- Wskaźniki, które da się w API BDL znaleźć po nazwie (np. stopa bezrobocia rejestrowanego), można wyszukać przy pomocy funkcji `znajdz_wskazniki_bdl()`.
    - W przypadku wskaźników, których wartości GUS raportuje z częstotliwością większą niż roczna (kwartalną, miesięczną), nazwa wskaźnika opisywana jest w zwracanych wynikach dwoma kolumnami: `n1` opisuje okres sprawozdawczy w ramach roku (kwartał lub miesiąc), a `n2` samą nazwę wskaźnika.
- W przypadku wskaźników, których nie da się w API BDL wyszukać po nazwie (np. przeciętne miesięczne wynagrodzenia brutto), trzeba samodzielnie (ręcznie) zidentyfikować ich `id` w API BDL, a następnie użyć funkcji `wskaznik_bdl()`, aby pobrać informacje o danym wskaźniku zwrócone w analogicznej formie jak ta, w jakiej zwraca informacje o wyszukanych wskaźnikach funkcja `znajdz_wskazniki_bdl()`.
    - Funkcja `wskaznik_bdl()` umożliwia przy tym nadpisanie nieinformatywnej nazwy wskaźnika pobranej z API BDL (i zwracaną w kolumnie `n1`) nazwą, której chcielibyśmy używać na potrzeby dalszych operacji na danych.

Rozróżnienie wskaźników jednego i drugiego rodzaju w praktyce polega na sprawdzeniu, czy dany wskaźnik daje się znaleźć po nazwie, co do której spodziewamy się, że ją ma – jeśli się to nie udaje, najprawdopodobniej jest to wskaźnik drugiego rodzaju.

Przykładowe użycia:

```
library(MLASZdane)
znajdz_wskazniki_bdl("stopa bezrobocia rejestrowanego")
wskaznik_bdl(64428, "przeciętne miesięczne wynagrodzenia brutto")
```

3.3.2. Pobieranie zestawień wartości wskaźników

Do pobierania zestawień wartości wskaźników z API BDL służy funkcja `pobierz_dane_bdl()`, która przyjmuje następujące argumenty:

- **wskazniki** – obiekt zwrócony przez funkcję `znajdz_wskazniki_bdl()` lub `wskaznik_bdl()` (lub dowolne połączenie wyników wywołania tych funkcji przy pomocy funkcji `rbind()` lub `bind_rows()`);
- **lata** – lata, dla których mają zostać pobrane wartości wskaźników (można podać wektor liczb, aby pobrać wartości z wielu lat);

- **poziom** – poziom agregacji, na którym mają zostać pobrane wartości wskaźników: "makroregiony", "województwa", "regiony", "podregiony", "powiaty" lub "gminy" (domyślnie "powiaty").

Funkcja zwraca ramkę danych o kolumnach:

- **idWsk** – id wskaźnika w API BDL;
- **subjectId** – id tematu (grupy wskaźników) w API BDL;
- **n1** i ew. **n2** - kolumny opisujące nazwy wskaźników w API BDL;
- **level** – najniższy poziom NTS, na którym raportowany jest wskaźnik;
- **measureUnitId**, **measureUnitName** – id i nazwa jednostki, w jakiej wyrażone są wartości wskaźnika;
- **idJst** – kod NTS jednostki terytorialnej, do której odnosi się wartość wskaźnika;
- **name** – nazwa jednostki terytorialnej, do której odnosi się wartość wskaźnika;
- **year** – rok, do którego odnosi się wartość wskaźnika;
- **val** – wartość wskaźnika;
- **attrId** – identyfikator atrybutu (co do zasady nieistotny);
- **teryt** – kod TERYT przeliczony na podstawie idJst.

Przykładowe użycia:

Uwaga! API BDL ma bardzo niskie limity na dopuszczalną w przedziale czasu liczbę zapytań, więc zapytanie takie jak poniżej można w praktyce wykonać tylko raz na 15 minut. W związku z tym zasadne jest zrobienie tego raz i zapisanie wyników lokalnie do wykorzystania w przyszłości (co w kodzie poniżej realizuje wywołanie funkcji `save()`; do wczytania obiektów zapisanych w pliku formatu .RData służy funkcja `load()`).

```
library(dplyr)
library(MLASZdane)
wskaznikiBdl =
  bind_rows(znajdz_wskazniki_bdl("stopa bezrobocia
rejestrowanego") %>%
              pobierz_dane_bdl(2017:2018, "powiaty"),
            wskaznik_bdl(64428, "przeciętne miesięczne
wynagrodzenia brutto") %>%
              pobierz_dane_bdl(2017, "powiaty"))
save(wskaznikiBdl, file = "wskazniki_BDL.RData")
```

3.3.3. Przekształcanie zestawień pobranych z API BDL na zestawienia wskaźników wykorzystywanych w monitorowaniu losów absolwentów

Przekształcenie zestawień wskaźników pobranych z API BDL na zestawienia wskaźników wykorzystywanych w monitorowaniu losów absolwentów polega na zmianie formy zestawienia z *długiej* (jedna JST-wiele wierszy) na *szeroką* (jedna JST-jeden wiersz) oraz przypisaniu kolumnom takiego zestawienia w formie *szerokiej* adekwatnych nazw. Przekształcenia te wykonuje funkcja `przeksztalc_dane_bdl()`, przy czym w obecnej postaci ma ona kilka ważnych ograniczeń:

- nazwy wszystkich wskaźników zawierające ciąg znaków „bezrobocia” traktowane są jako jeden wskaźnik, w nazwach wynikowych zmiennych opisywany jako „bezrobocie”;
- nazwy wszystkich wskaźników zawierające ciąg znaków „wynagrodzenia” traktowane są jako jeden wskaźnik, w nazwach wynikowych zmiennych opisywany jako „sr_wynagrodzenia”;
- nazwy wszystkich pozostałych wskaźników pozostają bez zmian, z tym że znaki spacji zamieniane są w nich na znak „_”.

Nie ma gwarancji, że ww. reguły nie ulegną zmianie w przyszłości!

Nazwy zmiennych zawierających wartości wskaźników w zwracanym zestawieniu są postaci: `jst_wskaznik_[[:digit:]]+[mr]`, gdzie:

- `wskaznik` to nazwa wskaźnika (p. wyżej);
- `[[:digit:]]+` to liczba (miesięcy lub lat od momentu planowego ukończenia szkoły przez uczniów rocznika podanego w wywołaniu funkcji argumentem rocznik);
- `[mr]+` to litera "m" lub litera "r" w zależności od tego, czy dany wskaźnik raportowany jest przez GUS z częstotliwością miesięczną, czy roczną.

Jeśli wszystkie wskaźniki w zestawieniu pobranym z API BDL przekazanym do funkcji są określone na tym samym poziomie agregacji, to w nazwach zmiennych ciąg znaków „jst” jest zamieniany na nazwę danego poziomu (np. "powiat").

Przy wywołaniu funkcji konieczne jest podanie rocznika, w odniesieniu do którego przygotowane zostaną nazwy zmiennych (przy czym funkcja arbitralnie zakłada, że absolwenci powinni planowo kończyć szkołę w czerwcu).

Przy pomocy opcjonalnego argumentu `prefiks` możliwe jest też dopisanie do nazw zmiennych identyfikujących w wynikowym zestawieniu JST (`teryt` i `nazwaJst`) prefiksu, tak aby odpowiadały one nazwom zmiennych w zbiorze, do którego ma być przyłączane zestawienie.

```
Przykładowe użycie:
```

```
library(MLASZdane)
load("wskazniki_BDL.RData")
przeksztalc_dane_bdl(wskaznikiBdl, 2017)
```

3.4. Wykorzystanie pakietu *MLASZdane* do przygotowania zbiorów wskaźników indywidualnych absolwentów

3.4.1. Obliczenie wskaźników na poziomie indywidualnym na podstawie zbiorów z pierwszej rundy monitoringu

Aby przygotować zbiór danych ze wskaźnikami opisującymi sytuację edukacyjno-zawodową poszczególnych absolwentów na podstawie wyników pierwszej rundy monitoringu, należy użyć funkcji `oblicz_wskazniki_ind_1rm()`.

Zakładając, że wcześniej wykonane zostały czynności opisane w rozdziale 3.2.1 i przetworzone zbiory danych z pierwszej rundy monitoringu (z zaimputowanymi brakami danych czasu rozpoczęcia i zakończenia *epizodów*) znajdują się w obiekcie `dane1RM`, wystarczy wywołać kod:

```
wskaznikiInd = oblicz_wskazniki_ind_1rm(dane1RM)
```

3.5. Wykorzystanie pakietu *MLASZdane* do przygotowania zbiorów wskaźników szkół

3.5.1. Obliczenie wskaźników na poziomie szkół i grup porównawczych na podstawie zbiorów z pierwszej rundy monitoringu

Aby obliczyć wskaźniki na poziomie szkół, oprócz zbioru danych ze wskaźnikami na poziomie indywidualnym potrzebny jest również zbiór ze wskaźnikami średnich wynagrodzeń i bezrobocia w powiatach pobranych z Banku Danych Lokalnych GUS. Aby przyłączyć pobrane zgodnie z wcześniejszymi instrukcjami wskaźniki z BDL do zbioru wskaźników indywidualnych, należy uruchomić kod:

```
load("wskazniki_BDL.RData")
wskaznikiBdl = przeksztalc_dane_bdl(wskaznikiBdl, 2017, "SZK_")
wskaznikiInd = left_join(wskaznikiInd, wskaznikiBdl)
```

Teraz można przystąpić do przygotowania zestawień wskaźników zagregowanych na potrzeby późniejszego generowania raportów szkół z wykorzystaniem pakietu *MLASZraporty*:

- na poziomie szkół,
- na poziomie typów szkół, które przy tworzeniu raportów posłużą jako punkt odniesienia dla ww. wskaźników na poziomie szkół.

Służą do tego funkcje, odpowiednio `agreguj_wskazniki_szk()` i `agreguj_wskazniki_typ_szk()`. Przygotowane przy ich pomocy zbiory można następnie zapisać, w celu późniejszego wykorzystania do przygotowania raportów szkół.

```
wskaznikiSzk = agreguj_wskazniki_szk(wskaznikiInd)
wskaznikiTypSzk = agreguj_wskazniki_typ_szk(wskaznikiInd)
save(wskaznikiSzk, wskaznikiTypSzk, file =
"wskazniki_szkol.RData")
```

W perspektywie tworzenia raportów na podstawie przygotowanego w wyżej opisany sposób zbioru danych należy jeszcze zwrócić uwagę, że szablon 'raport_szkoly.Rmd' zaimplementowany w pakiecie *MLASZraporty* wymaga dołączenia do zbioru wskaźników na poziomie szkół trzech dodatkowych zmiennych, których nie daje się wygenerować na podstawie zbiorów z wynikami sondaży z pierwszej rundy monitoringu (konieczne jest odwołanie się w tym celu do danych z operatu losowania próby do badania):

- **SZK_nazwa** - nazwa szkoły,
- **SZK_adres** - adres szkoły,
- **SZK_l_uczn_pop** - liczba uczniów w szkole należących do badanej populacji (w odróżnieniu zarówno od liczby uczniów wylosowanych do badania, jak i od liczby uczniów, których udało się zbadać).

3.5.2. Anonimizacja wskaźników na poziomie szkół

Zbiory danych zawierający wskaźniki obliczone na poziomie szkół, zwłaszcza w sytuacji, gdy zostały one obliczone w oparciu o dane zbierane przy pomocy sondażu (co wiąże się zwykle z niemożliwością uzyskania informacji o znacznej części absolwentów), może zawierać wartości wskaźników, które zostały obliczone na podstawie bardzo niewielkiej liczby absolwentów. Stanowi to zagrożenie dla anonimowości badanych, gdyż wartość wskaźnika na poziomie szkoły może być tu wykorzystana do wnioskowania (z pewnością lub z tylko niewielkim błędem) o wartości wskaźnika z poziomu indywidualnego. Połączenie ze sobą takich informacji z kilku wskaźników może, w pewnych okolicznościach (np. w sytuacji, gdy wskaźniki interpretowane są przez osoby posiadające znaczną wiedzę o badanych pochodzącą ze źródeł innych niż badanie absolwentów, jak np. nauczyciele, czy dyrektor szkoły), prowadzić do możliwości określenia prawdopodobnych odpowiedzi udzielonych przez konkretne osoby.

W celu uniknięcia takiego ryzyka przeprowadza się anonimizację zbiorów wskaźników na poziomie szkół, która polega na zastąpieniu wartości wskaźników, które zostały obliczone na podstawie mniejszej niż zadany próg liczby absolwentów kodami braku danych. Aby dokonać takiej anonimizacji, należy na obiekcie ze wskaźnikami z poziomu szkół wywołać funkcję `anonimizuj_wskazniki()`, jako drugi argument podając liczbę absolwentów (danej szkoły), poniżej której ma być przeprowadzona anonimizacja wartości wskaźnika (dot. tej szkoły):

```
wskaznikiSzk = anonimizuj_wskazniki(wskaznikiSzk, 10)
```

3.5.3. Eksport zbiorów wskaźników na poziomie zagregowanym

Przy pomocy funkcji `splaszcz_wskazniki_zagregowane()` możliwe jest przekształcenie przygotowanych zbiorów wskaźników na poziomie zagregowanym do

postaci płaskich ramek danych (tzn. niezawierających kolumn-list), co umożliwia zapisanie ich w formacie SPSS lub Staty:

```
wskaznikiSzkEksport = splaszcz_wskazniki_zagregowane(wskaznikiSzk)
library(haven)
write_dta(wskaznikiSzkEksport, "wskazniki_szkol.dta")
write_sav(wskaznikiSzkEksport, "wskazniki_szkol.sav")
```

3.6. Wykoszystanie pakietu *MLASZraporty* do generowania raportów szkół

3.6.1. Użycie funkcji `generuj_raporty()`

Do generowania raportów służy funkcja `generuj_raporty()`. Jej typowe wywołanie wygląda następująco:

```
library(MLASZraporty)
generuj_raporty(szablon = 'raport_szkoly.Rmd',
                wskazniki = wskaznikiSzk,
                wskaznikiGrPor = wskaznikiTypSzk,
                kolumnaNazwaPliku = SZK_kod,
                parametry = list(typDokumentu = "pdf",
                                 progLiczebnosci = 10,
                                 rocznik = 2017,
                                 wyrownanieTabWykr = "center"))
```

Poszczególne argumenty opisują:

- `szablon` plik szablonu raportu, który ma zostać wykorzystany,
- `wskazniki` obiekt (ramka danych) zawierająca wartości wskaźników: wiersze reprezentują grupy, dla których mają zostać wygenerowane raporty (np. szkoły), kolumny zawierają poszczególne wskaźniki;
  - w wywołaniu powyżej wykorzystywany jest obiekt `wskaznikiSzk`, stanowiący część pakietu *MLASZraporty*, który zawiera przykładowe dane kompatybilne z szablonem *raport_szkoly.Rmd*;
  - w praktycznych zastosowaniach zwykle wykorzystywany będzie obiekt wskaźników na poziomie zagregowanym (np. szkół) przygotowany przy pomocy pakietu *MLASZdane*;
- `wskaznikiGrPor` obiekt (ramka danych) zawierająca wartości wskaźników w grupach porównawczych: wiersze reprezentują grupy porównawcze, kolumny zawierają poszczególne wskaźniki; zwykle struktura tego obiektu jest niemal identyczna (z dokładnością do tego, co reprezentują wiersze) do obiektu przekazywanego argumentem `wskazniki`;
  - sposób wyboru odpowiedniej grupy porównawczej do wykorzystania w konkretnym raporcie jest zakodowany w pliku z szablonem raportu;

- w wywołaniu powyżej wykorzystywany jest obiekt `wskaznikiTypSzk`, stanowiący część pakietu *MLASZraporty*, który zawiera przykładowe dane kompatybilne z szablonem *raport_szkoly.Rmd*;

- w praktycznych zastosowaniach zwykle wykorzystywany będzie obiekt wskaźników na poziomie zagregowanym (np. typów szkół) przygotowany przy pomocy pakietu *MLASZdane*;

- `kolumnaNazwaPliku` nazwa kolumny w obiekcie zawierającym wartości wskaźników, która zostanie wykorzystana do nadania nazw plikom raportów;

  - tego argumentu można nie podawać – pliki raportów będą wtedy mieć nazwy *raportNR*, gdzie *NR* to numer wiersza w ramce danych przekazanej argumentem `wskazniki`;

- `parametry` lista dodatkowych parametrów, niezbędnych do wygenerowania raportów na podstawie szablonu; może być specyficzna dla szablonu; najczęściej występujące parametry, które trzeba podać to:

  - `typDokumentu` „pdf” lub „html”;

  - `progLiczebnosci` próg liczby badanych, poniżej której wartości wskaźnika nie zostaną pokazane w raporcie (zamiast tego wygenerowana zostanie informacja o zbyt małej liczbie absolwentów);

  - `rocznik` rok ukończenia szkoły przez absolwentów;

  - `wyrownanieTabWykr` wyrównanie (w poziomie) tabel i wykresów: „left”, „right” lub „center”;

Opisane powyżej dodatkowe parametry – z wyjątkiem `typDokumentu` – mogą być też przekazane jako kolumny ramki danych ze wskaźnikami grup (tj. ramki danych przekazywanej argumentem `wskazniki`) - wtedy nie muszą być już wpisywane jako elementy argumentu `parametry`.

Raporty zostaną utworzone w aktywnym folderze. Aby sprawdzić, jaki to folder, można użyć funkcji `getwd()` i ew. funkcji `setwd()`, aby go zmienić.

4. Testy oprogramowania

4.1. Wprowadzenie

W celu zapewnienia wysokiej jakości opracowanych programów oraz dowiedzenia, że poprawnie realizuje ono założone funkcjonalności, przeprowadzone zostały w ramach projektu różnorodne testy oprogramowania. Podstawową zastosowaną formą testów są tzw. testy wysokiego poziomu, w czasie których sprawdzano zdolność przygotowanych programów do bezbłędnego i poprawnego zrealizowania procesu wytworzenia raportów wszystkich szkół zbadanych w ramach pierwszej rundy monitoringu losów absolwentów,

począwszy od etapu przygotowania do analiz zbiorów danych opisujących wyniki badań sondażowych absolwentów. Oprócz nich przygotowane zostały również rozwiązania zapewniające kontrolę poprawności działania poszczególnych funkcji pakietu w różnych wariantach ich użycia (tzw. testy jednostkowe) oraz kontrolowanie, czy zmiany wprowadzane w kolejnych wersjach pakietów nie wprowadzają (nieintencjonalnych) błędów w działaniu funkcjonalności, które wcześniej działały poprawnie (tzw. testy regresji). Wykorzystane zostały przy tym nowoczesne rozwiązania wspierające pracę z kodem źródłowym i rozwój aplikacji, aby zapewnić możliwość automatycznego przeprowadzania testów (również w przyszłości) i raportowania stopnia pokrycia kodu testami.

4.2. Infrastruktura wykorzystywana do przeprowadzenia testów

Testy pakietów *MLASZdane* i *MLASZraporty* przygotowane zostały w ten sposób, aby stanowiły one integralną część tych programów, a ich przeprowadzenie było możliwe przez każdego z ich użytkowników[1]. W tym celu wykorzystany został pakiet *testthat* środowiska R, przy pomocy którego dokonano implementacji testów. W obu przygotowanych w ramach projektu pakietach testy znajdują się w podkatalogu *tests/testthat* katalogu głównego repozytorium zawierającego kod źródłowy pakietu.

Na potrzeby prowadzenia w okresie projektu, jak również w przyszłości, testów regresji dla obu pakietów skonfigurowano też połączone z repozytoriami kodu źródłowego rozwiązania zapewniające tzw. *ciągłą integrację* (ang. *continuous integration*), tj. uruchamianie testów pakietu po zapisaniu w repozytorium kodu każdej nowej wersji pakietu. Po przeprowadzeniu takich testów każdorazowo generowany jest raport opisujący stopień pokrycia kodu testami, co pozwala na łatwe zidentyfikowanie funkcjonalności, które nie są jeszcze testowane i odpowiednie uzupełnienie testów. W tym celu jako repozytorium kodu wykorzystano platformę *GitHub* (https://github.com), a jako narzędzie ciągłej integracji platformę *Travis CI* (https://travis-ci.org). Korzystanie z obu platform jest w odniesieniu do publicznie dostępnych repozytoriów darmowe.

Repozytoria z kodem źródłowym pakietów oraz raportami dotyczącymi pokrycia są dostępne pod adresami internetowymi:

- pakiet *MLASZdane*:
  - *https://github.com/tzoltak/MLASZdane*
  - *https://codecov.io/gh/tzoltak/MLASZdane*
- pakiet *MLASZraporty*:

[1] W przypadku pakietu *MLASZdane* do przeprowadzenia testów konieczne jest posiadanie plików ze zbiorami danych opisującymi wyniki badań sondażowych zrealizowanych w ramach pilotażowej i w ramach pierwszej rundy monitoringu, które, ze względu na ochronę anonimowości badanych, nie zostały umieszczone w publicznie dostępnym repozytorium z kodem pakietu.

- *https://github.com/tzoltak/MLASZraporty*
- *https://codecov.io/gh/tzoltak/MLASZraporty*

4.3. Poziom pokrycia testami

Kod źródłowy pakietu *MLASZdane* w wersji 0.3.2 jest pokryty testami w 96,5%, przy czym dla wszystkich funkcji z wyjątkiem `anonimizuj_wskazniki()` (90,9%), `koryguj_statusy()` (89,1%) i `spłaszcz_wskazniki_zagregowane()` (90,6%) pokrycie przekracza 99%. Każda z funkcjonalności pakietu została w procesie testowania przynajmniej raz uruchomiona i skorzystanie z niej nie spowodowało błędu.

Kod źródłowy pakietu *MLASZraporty* w wersji 0.1.3 jest pokryty testami w 98,5%, przy czym dla wszystkich funkcji z wyjątkiem `generuj_raporty()` (95,1%), `bez_naglowka_1kolumny()` (80%) i `dodaj_wiersz_sumy()` (94,1%) pokrycie wynosi 100%. Każda z funkcjonalności pakietu została w procesie testowania przynajmniej raz uruchomiona i skorzystanie z niej nie spowodowało błędu.

4.4. Testy wysokiego poziomu

Testy wysokiego poziomu objęły kontrolę poprawności współdziałania ze sobą różnych funkcji w ramach każdego z dwóch wytworzonych pakietów oraz obu tych pakietów między sobą. Do ich przeprowadzenia wykorzystywane zostały pełne zbiory danych pochodzące z badań zrealizowanych w ramach pilotażowej i pierwszej rundy monitoringu. Testy przeprowadzono poprzez manualne uruchamianie funkcji pakietów, realizując typowy schemat ich wykorzystania opisany w rozdziale 3. Testy przeprowadzono w systemie Microsoft Windows 10, korzystając ze środowiska R w wersji 3.5.3. Objęły one:

- **dla rundy pilotażowej:** przejście procedury przygotowania zbiorów danych *epizodów* (w postaci *długiej*) i zbioru *osobo-miesięcy*, w tym imputacji czasów rozpoczęcia i zakończenia *epizodów*, na podstawie zbioru z wynikami sondażu absolwentów;
- **dla pierwszej rundy:** przejście kolejno siedmiu procedur, prowadzących do wytworzenia raportów szkół, przy czym wynik działania poprzednich wykorzystywany był jako dane wejściowe dla dalszych z nich:
  1. przygotowanie zbiorów danych *epizodów* (w postaci *długiej*) i zbioru *osobo-miesięcy*, w tym imputacji czasów rozpoczęcia i zakończenia *epizodów*, na podstawie zbioru z wynikami sondażu ;
  2. obliczenie wskaźników indywidualnych na podstawie zbioru danych *epizodów*;
  3. pobranie z BDL GUS wskaźników dotyczących bezrobocia i przeciętnych wynagrodzeń w powiatach w latach 2017-2018 i zapisanie pobranego zestawienia;

4. przekształcenie zbioru danych ze wskaźnikami pobranymi z BDL GUS (przy pomocy funkcji `przeksztalc_dane_bdl()`),

5. obliczenie wskaźników na poziomie zagregowanym na podstawie zbioru wskaźników na poziomie indywidualnym; agregacja przeprowadzana jest kolejno (niezależnie od siebie) na poziomie: szkół, typów szkół, branży w ramach szkoły, branży w ramach typu szkoły,

6. *spłaszczenie* zbioru wskaźników na poziomie zagregowanym w celu przygotowania go do eksportu;

7. wygenerowanie raportów szkół na podstawie zbioru danych ze wskaźnikami zagregowanymi na poziomie szkół i zbioru danych ze wskaźnikami zagregowanymi na poziomie typów szkół (dla każdej szkoły wygenerowano raport w dwóch formatach: pliku PDF i pliku HTML).

Procedury 1.-6. realizowane były przy pomocy pakietu *MLASZdane*, a ostatnia przy pomocy pakietu *MLASZraporty*.

Łącznie wygenerowano raporty 1586 raportów, po 793 w każdym z dwóch formatów (PDF lub HTML). Ich zawartość poddana została następnie sprawdzeniu w celu skontrolowania poprawności zapisanych w nich wyników oraz zgodności uzyskanego formatowania tekstu, tabel i wykresów z przyjętymi założeniami.

Przeprowadzone testy potwierdziły poprawność działania pakietów *MLASZdane* i *MLASZraporty* jako narzędzi pozwalających wygenerować automatycznie raporty szkół zawodowych na podstawie danych sondażowych zebranych w ramach pierwszej rundy monitoringu losów absolwentów.

4.5. Testy jednostkowe

Procedury opisanych powyżej testów wysokiego poziomu obejmują wykonanie niemal wszystkich funkcji obydwu opracowanych pakietów z wykorzystaniem szerokiego zakresu danych wejściowych. W przypadku niemal wszystkich tych funkcji zapewnia to pokrycie testami powyżej 99% linii kodu (wiąże się to również z faktem, że większość funkcji obu pakietów – w szczególności funkcje odpowiedzialne za obliczanie wskaźników – jest wąsko wyspecjalizowana i nie ma zróżnicowanych ścieżek wykonania). W związku z tym przeprowadzane testy wysokiego poziomu pełnią również funkcję testów jednostkowych. Jak zostało to już wspomniane, rozwiązania służące automatycznemu przeprowadzaniu tych testów zostały zaimplementowane jako integralny element repozytorium kodu źródłowego obu przygotowanych pakietów. Testy przeprowadzane automatycznie zasadniczo replikują opisane wyżej testy wysokiego poziomu, z kilkoma wymienionymi poniżej różnicami:

- Testy przeprowadzane automatycznie przeprowadzane są oddzielnie dla każdego z pakietów.

- Do testów pakietu *MLASZraporty* wykorzystywane są przykładowe zbiory danych stanowiące zanonimizowane **podzbiory**:

- zbioru wskaźników zagregowanych na poziomie szkół,
- zbioru wskaźników zagregowanych na poziomie typów szkół,

przygotowanych przy pomocy pakietu *MLASZdane* w wersji 0.3.2 na podstawie danych z wynikami badań sondażowych z pierwszej rundy monitoringu losów absolwentów. Zbiory te stanowią element repozytorium kodu pakietu *MLASZraporty* i **nie są** automatycznie aktualizowane po wprowadzeniu zmian w kodzie pakietu *MLASZdane*.

- Testy przeprowadzane automatycznie obejmują jedynie kontrolę, że wywołanie funkcji nie powoduje błędów programu oraz sprawdzenie poprawności ogólnej struktury zwracanych obiektów lub generowanych plików. Nie obejmują one jednak szczegółowej kontroli poprawności zwracanych wartości.
- Aby zapewnić pokrycie fragmentów kodu niepokrytych w ramach realizacji opisanej wcześniej procedury testów wysokiego poziomu, dla niektórych funkcji napisane zostały dodatkowe testy jednostkowe:
  - W pakiecie *MLASZdane* - dla funkcji pomocniczych `przywroc_etykiety()` oraz `sprawdz_nazwy()` przygotowane zostały dodatkowe testy jednostkowe, sprawdzające poprawność ich zachowania w sytuacjach, w których ich wywołanie powinno zakończyć się błędem lub wygenerować komunikat ostrzegawczy.
  - W pakiecie *MLASZraporty* – przygotowane zostały testy jednostkowe, sprawdzające poprawność zachowania się funkcji `generuj_raporty()` w nietypowych sytuacjach, w których jej wywołanie powinno zakończyć się błędem lub wygenerować komunikat ostrzegawczy.

4.6. Testy regresji

Wszystkie przygotowane testy jednostkowe uruchamiane są automatycznie po dokonaniu każdej zmiany w kodzie źródłowym pakietu (dokładnie przy zatwierdzeniu zmian w repozytorium), co spełnia funkcję testów regresji, tzn. pozwala na natychmiastowe stwierdzenie, że zmiana dokonana w pakiecie spowodowała błąd w funkcjonalnościach, które dotychczas działały poprawnie.