Imputacja danych w R

Litwiński Michał

1 Metody imputacji i pakiety służące imputowaniu danych w R

W tabeli 1 przedstawiono funkcje, które można wykorzystać podczas imputacji danych przy pomocy wybranej metody: imputacja regresyjna, imputacja z wykorzystaniem średniej, predykcyjne dopasowanie według średniej, imputacja z wykorzystaniem innej zmiennej, imputacja metodą najbliższego sąsiada oraz imputacja typu hot-deck.

Imputacja regresyjna polega na zastępowaniu danych brakujących o wartości oszacowane przy pomocy modelu regresji. W dużej mierze na tej metodzie opiera się imputacja wielokrotna, którą opisano w podrozdziale 3.1. Zarówno pakiet *mi* jak i *mice* w zależności od specyfikacji wykorzystują przy imputacji model regresji.

Imputacja z wykorzystaniem średniej – ta metoda polega na zastąpieniu wartości brakujących średnią wartością zmiennej obliczoną dla wszystkich obserwacji, dla których dane istnieją lub dla części obserwacji wyodrębnionych na podstawie danej cechy [Szymkowiak 2009].

Predykcyjne dopasowanie według średniej (ang. *predictive mean matching*) – w tej metodzie dana zmienna y jest szacowana na podstawie zbioru zmiennych x_i (model regresji, klasyczna normalna regresja liniowa). Następnie każdą jednostkę z brakiem danych (nierespondenta) łączy się z jednostką, dla której zaobserwowano wartość zmiennej (z respondentem) na zasadzie zbieżności wartości y. Nierespondentowi przypisuje się wartość zmiennej zaobserwowaną dla respondenta o najbliższej nierespondentowi wartości y [Little 1988].

Imputacja metodą najbliższego sąsiada polega na znalezieniu dawcy wartości danej zmiennej y spośród jednostek, dla których zaobserwowano wartość zmiennej. Dawca jest określany na podstawie wartości innych cech (x_i) na zasadzie minimalizacji odległości między wartościami tych cech dla dawcy i biorcy.

Imputacja typu hot-deck – w tej metodzie imputowana wartość zmiennej pochodzi od losowo wybranego dawcy, dla którego zaobserwowano wartość zmiennej. Losowanie jest dokonywane spośród

wszystkich obiektów z kompletnymi danymi lub spośród określonej klasy imputacyjnej (grupy wyodrębnionej na podstawie innych cech) [Szymkowiak 2009].

W opracowaniu wykorzystano dane z bazy Diagnoza społeczna. W celu ułatwienia i przyspieszenia pracy na zbiorze, ograniczono go, do 50 obserwacji oraz wybrano zmienne, które przedstawia zestawienie 1. Przykłady zastosowań poszczególnych funkcji przedstawiono w plikach *RMarkdown*. Do każdego rozdziału opracowania odnosi się oddzielny plik.

Zestawienie 1: Zmienne z badania Diagnoza społeczna wykorzystane w analizie pakietów dotyczących imputacji danych

```
[1] "L_OSOB_2009"        "L_OSOB_2011"        "L.OSOB_2013"        "WAGA_GD_2000"
[5] "WAGA_GD_2013"       "KLASA_MIEJSCOWOSCI" "WOJEWODZTWO"        "Podregion_58"
[9] "GF8_01"
```

Tabela 1: Funkcje dostępne w pakietach wg metod imputacji

Pakiet	**Imputacja regresyjna**	**Imputacja z wykorzystaniem średniej**	**Predykcyjne dopasowanie według średniej**	**Imputacja metodą k najbliższych sąsiadów**	**Imputacja typu hotdeck**
mi	*mi*, funkcje dla danej zmiennej		*mi.pmm*		
mice	*mice*, funkcje dla danej zmiennej	*mice.impute.mean*	*mice.impute.pmm*		
Amelia	*amelia*				
VIM	*irmi*, *regressionImp*	*initalise*		*kNN*	*hotdeck*
mix					
pan					
norm					
cat					
MImix					
robCompositisons					

Pakiet	Imputacja regresyjna	Imputacja z wykorzystaniem średniej	Predykcyjne dopasowanie według średniej	Imputacja metodą najbliższego sąsiada	Imputacja typu hot-deck
VIM					
yaImpute					
robCompositions					
rrcovNA					
impute					
miss-MDA					
mice					
Hmisc					
VIM					

Uwagi: w tabeli umieszczono nazwy funkcji (*kursywą*), które można wykorzystać do imputacji
Źródło: opracowanie własne na podstawie [Szymkowiak 2009].

2 Funkcje wykorzystywane do podsumowań braków danych

2.1 Tabelaryczne przedstawienie braków danych

Pakiet *mice* W pakiecie mice znajdują się funkcje, które w prosty sposób podsumowują braki danych dla danego zbioru danych. Ich opis oraz wynik działania przedstawia tabela 2.

Tabela 2: Funkcje służące podsumowaniu braków danych w ramach pakietu mice

Funkcja	Opis	Wynik dla zbioru danych
ccn	liczba wierszy bez braków danych	3
icn	liczba wierszy z przynajmniej jednym brakiem danych	47
cci	zwraca wartość logiczną dla każdego wiersza (TRUE = wiersz nie zawiera braków danych; FALSE = wiersz zawiera przynajmniej jedna wartość brakującą)	
cc	zwraca wiersze bez braków danych	

Uwagi: zbiór danych pochodzi z pliku DIAGNOZA (szczegółowe informacje w plikach RMarkdown.
Źródło: opracowanie własne na podstawie [van Buuren and Groothuis-Oudshoorn 2011].

W ramach pakietu *mice* dostępna jest również funkcja *md.pattern*, która w dosyć przejrzysty sposób prezentuje zbiór danych pod względem wartości brakujących. W zestawieniu 2 przedstawiono wynik zastosowania tej funkcji dla 5 zmiennych.

Z tabeli można odczytać, że w zbiorze znajdują się 23 wiersze bez braków danych, 13 wierszy, gdzie brakuje tylko wartości dla zmiennej L_OSOB_2009 oraz 14 takich, w których brakuje danych dla zmiennych L_OSOB_2009 i L_OSOB_2011. Ostatnia kolumna wskazuje, ile jest zmiennych z

brakującymi wartościami dla danej liczby wierszy, a ostatni wiersz wskazuje, ile braków danych zaobserwowano dla danej zmiennej. Na przecięciu ostatniej kolumny i ostatniego wiersza umieszczono sumę braków danych dla całego zbioru.

Zestawienie 2: Wynik zastosowania funkcji *md.pattern*

	numer_gd	nr_sztywny_15062013	L.OSOB_2013	L_OSOB_2011	L_OSOB_2009	
23	1	1	1	1	1	0
13	1	1	1	1	0	1
14	1	1	1	0	0	2
	0	0	0	14	27	41

Ciekawą funkcją, zwłaszcza w kontekście analizy relacji pomiędzy poszczególnymi zmiennymi ze zbioru jest *md.pairs*. Pozwala ona na sprawdzenie liczby wartości brakujących dla konkretnych par zmiennych. Wynik zastosowania tej funkcji zawiera zestawienie 3.

Zestawienie składa się z 4 części:

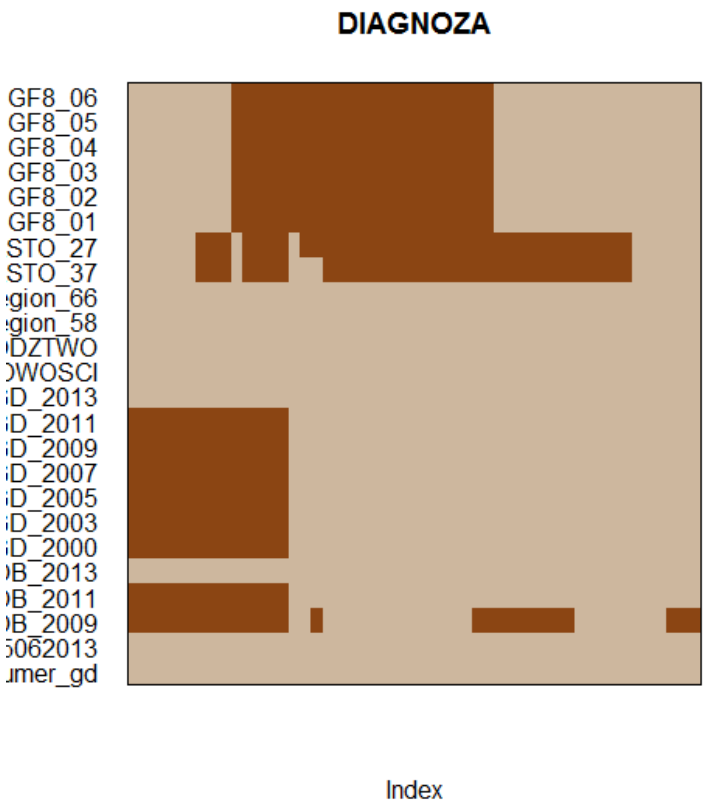
- **rr (*response-response*)** – wartości dla obydwu zmiennych są obserwowane (np. dla zmiennych L_OSOB_2009 i L_OSOB_2011 są 23 takie przypadki)
- **rm (*response-missing*)** – wartości dla zmiennej w wierszu są obserwowane, a wartości dla zmiennej w kolumnie - brakujące (np. dla zmiennych L_OSOB_2009 i L_OSOB_2011 są 23 takie przypadki)
- **mr (*missing-response*)** – wartości dla zmiennej w kolumnie są obserwowane, a wartości dla zmiennej w wierszu - brakujące (np. dla zmiennych L_OSOB_2009 i L_OSOB_2011 jest 13 takich przypadków)
- **mm (*missing-missing*)** – wartości dla obydwu zmiennych są brakujące (np. dla zmiennych L_OSOB_2009 i L_OSOB_2011 jest 14 takich przypadków)

Zestawienie 3: Wynik zastosowania funkcji *md.pairs*

```
$rr
                    numer_gd nr_sztywny_15062013 L_OSOB_2009 L_OSOB_2011 L.OSOB_2013
numer_gd                  50                  50          23          36          50
nr_sztywny_15062013       50                  50          23          36          50
L_OSOB_2009               23                  23          23          23          23
L_OSOB_2011               36                  36          23          36          36
L.OSOB_2013               50                  50          23          36          50

$rm
                    numer_gd nr_sztywny_15062013 L_OSOB_2009 L_OSOB_2011 L.OSOB_2013
numer_gd                   0                   0          27          14           0
nr_sztywny_15062013        0                   0          27          14           0
L_OSOB_2009                0                   0           0           0           0
L_OSOB_2011                0                   0          13           0           0
L.OSOB_2013                0                   0          27          14           0

$mr
                    numer_gd nr_sztywny_15062013 L_OSOB_2009 L_OSOB_2011 L.OSOB_2013
numer_gd                   0                   0           0           0           0
nr_sztywny_15062013        0                   0           0           0           0
L_OSOB_2009               27                  27           0          13          27
L_OSOB_2011               14                  14           0           0          14
L.OSOB_2013                0                   0           0           0           0

$mm
                    numer_gd nr_sztywny_15062013 L_OSOB_2009 L_OSOB_2011 L.OSOB_2013
numer_gd                   0                   0           0           0           0
nr_sztywny_15062013        0                   0           0           0           0
L_OSOB_2009                0                   0          27          14           0
L_OSOB_2011                0                   0          14          14           0
L.OSOB_2013                0                   0           0           0           0
```

2.2 Graficzne przedstawienie braków danych

Pakiet *mi* W ramach pakietu *mi* dostępna jest funkcja *missing.pattern.plot*, która pozwala graficznie przedstawić wzór, według ktorego układają się wartości brakujące. Na wykresie 1 ciemniejszym kolorem zaznaczono wartości brakujące.



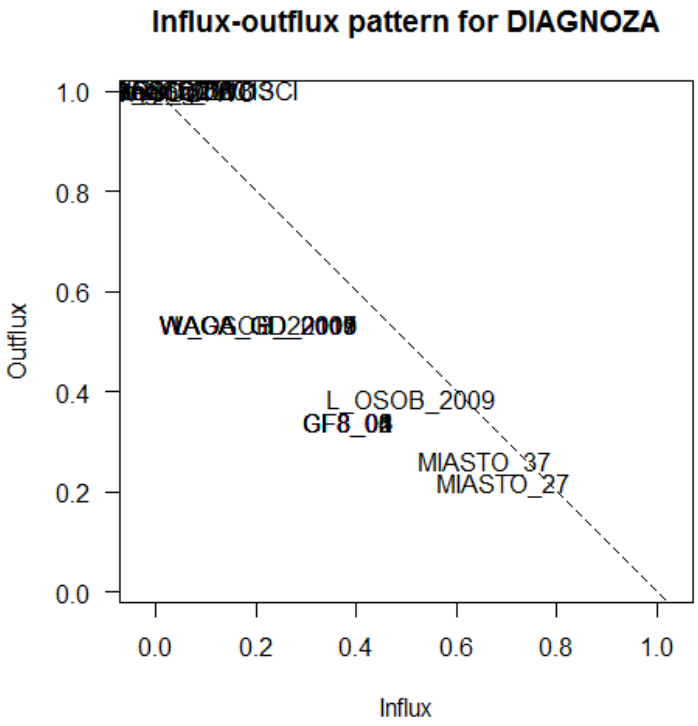
Wykres 1: Wykres wygenerowany za pomocą funkcji *missing.pattern.plot*

Pakiet *mice* W ramach pakietu dostępna jest bardzo ciekawa funkcja *fluxplot*, która pozwala określić wzór występowania wartości brakujących. Opiera się ona na wartościach *influx* i *outflux* [van Buuren 2012].

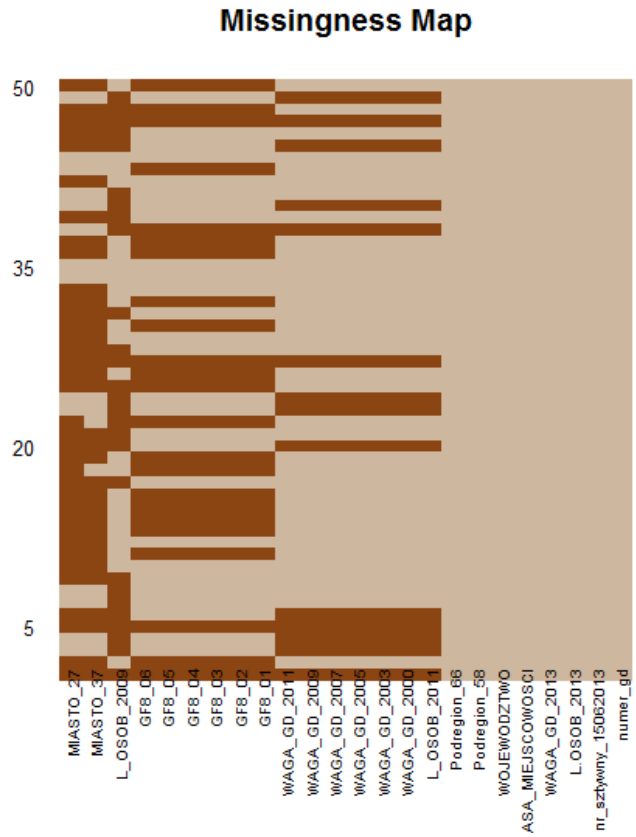
Influx to liczba par wartości zmiennych: x i y, w których dla x występuje brak danych, a dla y wartość zmiennej została zaobserwowana, podzielona przez całkowitą liczbę obserwowanych wartości. Dla zmiennej, w której nie występują braki danych *influx* przyjmuje wartość 0. Dla dwóch zmiennych z tym samym udziałem wartości brakujących, zmienna z wyższą wartością *influx* jest lepiej powiązana ze zmiennymi w pełni obserwowanymi, więc jest łatiejsza do imputacji.

Outflux to liczba par wartości zmiennych: x i y, w których dla y występuje brak danych, a dla x wartość zmiennej została zaobserwowana, podzielona przez całkowitą liczbę braków danych. Jest to wskaźnik potencjalnej przydatności zmiennej x do wykorzystania w modelu imputacji innych zmiennych (im wyższy wskaźnik, tym przydatność większa). *Outflux* zmiennej, w której nie występują braki danych przyjmuje wartość 1.

Pakiet *Amelia* W ramach pakietu dostępna jest funkcja *missmap*, która pozwala wygenerować mapę i określić sposób rozmieszczenia wartości brakujących. Na wykresie 3 wartości brakujące oznaczono ciemniejszym kolorem.



Wykres 2: Wykres wygenerowany za pomocą funkcji *fluxplot*



Wykres 3: Wykres wygenerowany za pomocą funkcji *missmap*

3 Metody wykorzystywane w poszczególnych pakietach

3.1 Imputacja wielokrotna

Imputacja wielokrotna polega na zastąpieniu każdej brakującej wartości danej zmiennej zestawem możliwych liczb, uwzględniającym niepewność dotyczącą prawdziwej wartości (zazwyczaj zestaw ten składa się z 3-10 wartości). Następnie imputowane zbiory danych są analizowane przy pomocy wybranej metody statystycznej (wykorzystuje się metody opracowane dla kompletnych danych), dzięki czemu możliwe jest połączenie tych zbiorów. W rezultacie powstaje jeden zestaw wartości imputowanych [Rubin 1987].

Przewagą imputacji wielokrotnej nad jednokrotną jest uwzględnienie zmienności pomiędzy poszczególnymi wersjami szeregów wartości imputowanych. Imputacja jednokrotna nie uwzględnia niepewności odnośnie prawdziwych wartości zmiennej. Ponadto, imputacja wielokrotna zachowuje najważniejsze cechy zbioru danych jako całości [Yuan 2000].

Imputacja wielokrotna opiera się na założeniu, że wartości brakujące są *missing at random*, co oznacza, że prawdopodobieństwo, że wystąpi brak danych może zależeć tylko od wartości zmiennych, dla których występuje pełna informacja, a nie od tych, dla których obserwuje się braki danych [Rubin 1987].

Mechanizm imputacji wielokrotnej opiera się na metodzie MCMC (*Markov chain Monte Carlo*) [Yuan 2000]. Jest to metoda wykorzystywana do generowania pseudolosowych zestawów liczb na podstawie wielowymiarowych i trudnych do kontrolowania rozkładów prawdopodobieństwa. Metoda ta posługuje się łańcuchami Markova [1]. Symulacje wartości brakujących są dokonywane na podstawie rozkładu Bayesa. Zastosowanie MCMC polega na konstrukcji łańcucha Markova na tyle długiego, by rozkład elementów ustabilizował się na danym poziomie. Następnie na podstawie tego rozkładu generowane są zestawy danych brakujących.

Zastosowanie metody wygląda następująco:

1. **Krok 1:** Na podstawie wyestymowanej średniej i macierzy kowariancji dla każdej wartości brakującej generowana jest wartość imputowana. Jeżeli:

   $Y_{i(mis)}$ – zmienna z wartościami brakującymi dla obserwacji i,

   $Y_{i(obs)}$ – zmienna z wartościami obserwowanymi dla obserwacji i,

   każdemu $Y_{i(mis)}$ przypisywana jest wartość $Y_{i(obs)}$ na podstawie warunkowego rozkładu $Y_{i(mis)}$.

2. **Krok 2:** Generowana jest średnia i macierz kowariancji dla populacji (na podstawie wartości wyznaczonych w poprzednim kroku). Te wartości są ponownie wykorzystywane w pierwszym

[1] Łańcuch Markova jest sekwencją zmiennych losowych, w której rozkład każdego elementu zależy od wartości poprzedniego

kroku [2].

Wartości początkowe dla obliczeń są wyznaczane na podstawie algorytmu EM (*expectation-maximization*). Alogorytm EM jest techniką, za pomocą której szuka się wartości maksymalnego prawdopodobieństwa (ang. *maximum likelihood*) dla modeli parametrycznych wykorzystywanych dla zbiorów z brakami danych [Little and Rubin 2002]. W kontekście wykorzystania MCMC zastosowanie EM polega na wyznaczeniu średnich i odchyleń standardowych na podstawie dostępnych danych dla zmiennych, dla których obserwujemy braki danych. Następnie na podstawie odchyleń standardowych wyznaczana jest macierz kowariancji [Schafer 1997].
Kroki imputacji wielokrotnej [Yuan 2000]:

1. **Wybór modelu łączącego zmienne, dla których występują braki danych ze zmiennymi, dla których mamy pełną informację**. W tym kroku określamy model, który pozwoli wygenerować wartości mające zastąpić braki danych (np. model liniowy lub logistyczny). Wybór następuje na podstawie charakteru zmiennych(np. zmienna ciągła/binarna, numeryczna/porządkowa). W specyfikacji modelu uwzględniamy zmienne, które są skorelowane ze zmienną, dla której brakuje danych i/lub zmienne, które są przyczyną braków danych).

2. **Wygenerowanie wartości, którymi można zastąpić brakujące** (możliwe zestawy danych dla każdej wartości brakującej). W tym kroku stosuje się k razy (zazwyczaj $k \in \langle 3;10 \rangle$) model wybrany w poprzednim kroku. Dana imputacja k jest poprzedzona wykonaniem iteracji na podstawie MCMC (mechanizm metody wyjaśniono powyżej).

3. **Przeprowadzenie analizy statystycznej określonego typu na tych zbiorach danych**. Analiza jest prowadzona tak, jakby braki danych nie występowały. Przykładowo, oblicza się średnią, wariancję lub przeprowadza analizę regresji.

4. **Połączenie wyników tych analiz, aby stworzyć jeden zestaw rezultatów**. W tym kroku oblicza się srednią i wariancję dla parametrów z poprzedniego kroku wyznaczonych dla poszczególnych zbiorów danych.

   K – liczba wygenerowanych zbiorów danych (liczba zestawów imputowanych wartości)

   k – numer określonego zbioru danych

   $\overline{S}$ – średnia parametrów wyliczonych w poprzednim kroku

   S_k – parametr obliczony dla danego zbioru danych (np. średnia, parametry równania regresji)

   T – wariancja

   $\overline{W}$ – średnia wariancja dla zbiorów danych

[2] Iteracje na podstawie tych dwóch kroków są wykonywane tak długo, dopóki rezultaty nie są wystarczająco wiarygodne dla imputacji wielokrotnej, tzn. rozkład elementów będzie stabilny.

W_k – wariancja dla danego zbioru danych

B – wariancja pomiędzy zbiorami danych

Średnią oblicza się w następujący sposób:

$\overline{S} = \frac{1}{k} \sum_{k=1}^{K} S_k$

Natomiast wariancja składa się z dwóch komponentów (wariancja w zbiorze danych i pomiędzy zbiorami danych):

$T = \overline{W} + (1 + K^{-1})B$, gdzie

$\overline{W} = K^{-1} \sum_{k=1}^{K} W_k$

$B = (K-1)^{-1} \sum_{k=1}^{K} (\hat{S_k} - \overline{S})^2$

Pakiet *mi* Pakiet *mi* umożliwia wgląd i ingerencję w proces imputacji danych, a także ocenę tego, czy wartości zostały wyestymowane we właściwy sposób. Pakiet daje możliwość wyboru modelu imputacji i transformacji tego modelu. Dostępne są również funkcje, służące tworzeniu wykresów sprawdzających dopasowanie wykorzystanych w imputacji rozkładów warunkowych, a także porównujących wartości rzeczywiste z wygenerowanymi.

Pakiet wykorzystuje algorytm oparty na równaniach łańcuchowych. Użytkownik określa rozkład warunkowy zmiennych z brakującymi wartościami oparty na pozostałych zmiennych. Następnie algorytm jest stosowany sekwencyjnie po zmiennych, aby wygenerować brakujące wartości zgodnie z określonym modelem.

Podstawą pakietu *mi* jest funkcja o tej samej nazwie. Pierwszym krokiem imputacji przy pomocy funkcji *mi* jest wygenerowanie pliku informacyjnego dla tej funkcji. Został on przedstawiony w zestawieniu 4.

mi.info(data)

Zestawienie 4: Plik informacyjny dla funkcji *mi*

	names	include	order	number.mis	all.mis	type	collinear
1	L_OSOB_2009	Yes	1	27	No	ordered-categorical	No
2	L_OSOB_2011	Yes	2	14	No	positive-continuous	No
3	L.OSOB_2013	Yes	NA	0	No	positive-continuous	No
4	WAGA_GD_2000	Yes	3	14	No	count	No
5	WAGA_GD_2013	Yes	NA	0	No	positive-continuous	No
6	KLASA_MIEJSCOWOSCI	Yes	NA	0	No	positive-continuous	No
7	WOJEWODZTWO	Yes	NA	0	No	positive-continuous	No
8	Podregion_58	Yes	NA	0	No	positive-continuous	No
9	GF8_01	Yes	4	23	No	binary	No

Uwzględnione informacje:

- names – nazwa zmiennej
- include – czy zmienna zostanie włączona do procedury imputacyjnej

- order – kolejność przy imputacji
- number.mis – liczba brakujących wartości
- all.mis – czy wszystkie wartości są brakujące
- type – typ zmiennej
- collinear – zmienne powiązane (ang. *collinear*)

Plik ten można modyfikować przy pomocy funkcji *mi.info.update*.

Tak przygotowany plik wykorzystuje się następnie we właściwej funkcji *mi*. Zastosowanie funkcji pozwala otrzymać macierz, którą zawiera zestawienie 5.

Zestawienie 5: Wynik obliczeń przy pomocy funkcji *mi*

```
Multiply imputed data set

Call:
 mi.default(data = object, info = info, n.imp = n.imp, n.iter = n.iter,
    R.hat = R.hat, max.minutes = max.minutes, rand.imp.method = rand.imp.method,
    run.past.convergence = run.past.convergence, seed = seed,
    check.coef.convergence = check.coef.convergence, add.noise = add.noise)

Number of multiple imputations:  3

Number and proportion of missing data per column:
               names                type number.mis proportion
1        L_OSOB_2009 ordered-categorical         27       0.54
2        L_OSOB_2011 positive-continuous         14       0.28
3        L.OSOB_2013 positive-continuous          0       0.00
4       WAGA_GD_2000         nonnegative         14       0.28
5       WAGA_GD_2013 positive-continuous          0       0.00
6 KLASA_MIEJSCOWOSCI positive-continuous          0       0.00
7        WOJEWODZTWO positive-continuous          0       0.00
8       Podregion_58 positive-continuous          0       0.00
9             GF8_01              binary         23       0.46

Total Cases: 50
Missing at least one item: 4
Complete cases: 8
```

Najważniejsze argumenty funkcji:

- object – Ramka danych lub obiekt klasy *mi*, zawierające braki danych (brak danych = NA)
- info – plik informacyjny (z funkcji *mi.info*)
- n.imp – liczba przeprowadzonych imputacji (liczba zestawów danych, które można imputować)
- n.iter – liczba iteracji (iteracje z wykorzystaniem metody MCMC, które służą wygenerowaniu danego zestawu wartości imputowanych)
- R.hat – wartość statystyki używanej jako kryterium konwergencji
- max.minutes – max. liczba minut na cały proces

- seed – liczba całkowita, będąca argumentem generatora liczb pseudolosowych

Informacje uwzględnione w wyniku:

- names – nazwa zmiennej
- type – typ zmiennej
- number.mis – liczba brakujących wartości
- proportion – udział brakujących wartości

Funkcja *mi* może się odnosić do określonego typu zmiennej. Obecnie obsługuje 10 typów, które zostały przedstawione w tabeli 3 wraz z odpowiadającymi im funkcjami. Modelem imputacji stosowanym w przypadku tych funkcji jest regresja liniowa.

Typ zmiennej można sprawdzić za pomocą funkcji *typecast(object)*.

Tabela 3: Funkcja *mi* dla określonych typów zmiennych

Typ zmiennej	**Objaśnienie**	**Funkcja**
binarna	zmienna przyjmuje dwie wartości (zazwyczaj: 0, 1)	mi.binary
ciągła	zmienna przyjmuje wartości ze zbioru liczb rzeczywistych	mi.continuous
typu *count*	zmienna przyjmująca wartości zgodnie z rozkładem Poissona	mi.count
stała	zmienna przyjmująca jedną wartość	mi.fixed
ciągła logarytmiczna	zmienna przyjmująca wartości zgodnie z funkcją logarytmiczną	mi.continuous
nieujemna	zmienna przyjmująca wartości ze zbioru liczb rzeczywistych, ale >= 0	mi.continuous
uporządkowana kategoryczna	zmienna przyjmująca od 3 do 5 kolejnych wartości	mi.polr
nieuporządkowana kategoryczna	zmienna przyjmująca więcej niż 2 wartości, które nie muszą być uporządkowane (faktor)	mi.categorical
dodatnia ciągła	zmienna przyjmująca wartości ze zbioru liczb rzeczywistych, ale > 0	mi.continuous
proporcjonalna	zmienna przyjmująca wartości z rozkładu β	mi.continuous

Źródło: opracowanie własne na podstawie [van Buuren and Groothuis-Oudshoorn 2011].

Sprawdzenie występujących typów zmiennej i funkcji im odpowiadających jest możliwe dzięki zastosowaniu funkcji *mi.types()*, a następnie *type.models("type")*.

Pakiet *mice* Pakiet ten wykorzystuje metodę imputacji wielokrotnej w oparciu o równania łańcuchowe (ang. MICE). Metoda ta jest znana również pod nazwą *fully conditional specification*. Polega ona na stworzeniu wielowymarowego modelu imputacji w oparciu o warunkowe gęstości dla każdej

zmiennej zawierającej braki danych. Na tej podstawie, w wyniku kolejnych iteracji powstają wartości uzupełniające (zazwyczaj wystarczającą liczbą iteracji jest 10-20).

Pakiet *mice* radzi sobie nie tylko z brakami danych o rozkładzie losowym, ale działa również w sytuacji, gdy brakujące wartości nie spełniają założenia losowości(ang. *missing not at random*) [van Buuren and Groothuis-Oudshoorn 2011, s. 15].

Podstawą pakietu jest funkcja *mice*, którą przedstawiono w zestawieniu 6.

Zestawienie 6: Wynik obliczeń przy pomocy funkcji *mice*

```
Multiply imputed data set
Call:
mice(data = DIAGNOZA[, c(3:6, 12:15, 19)], m = 10, method = c("pmm",
    "pmm", " ", "pmm", " ", " ", " ", " ", "pmm"), predictorMatrix = (1 -
    diag(1, ncol(DIAGNOZA[, c(3:6, 12:15, 19)]))), maxit = 5,
    printFlag = FALSE, seed = 100)
Number of multiple imputations:  10
Missing cells per column:
       L_OSOB_2009        L_OSOB_2011        L.OSOB_2013       WAGA_GD_2000
                27                 14                  0                 14
      WAGA_GD_2013 KLASA_MIEJSCOWOSCI        WOJEWODZTWO       Podregion_58
                 0                  0                  0                  0
            GF8_01
                23
Imputation methods:
       L_OSOB_2009        L_OSOB_2011        L.OSOB_2013       WAGA_GD_2000
            "pmm"              "pmm"                " "              "pmm"
      WAGA_GD_2013 KLASA_MIEJSCOWOSCI        WOJEWODZTWO       Podregion_58
               " "                " "                " "               " "
            GF8_01
             "pmm"
VisitSequence:
 L_OSOB_2009  L_OSOB_2011 WAGA_GD_2000       GF8_01
           1            2             4            9
PredictorMatrix:
                  L_OSOB_2009 L_OSOB_2011 L.OSOB_2013 WAGA_GD_2000 WAGA_GD_2013
L_OSOB_2009                 0           1           1             1             1
L_OSOB_2011                 1           0           1             1             1
L.OSOB_2013                 0           0           0             0             0
WAGA_GD_2000                1           1           1             0             1
WAGA_GD_2013                0           0           0             0             0
KLASA_MIEJSCOWOSCI          0           0           0             0             0
WOJEWODZTWO                 0           0           0             0             0
Podregion_58                0           0           0             0             0
GF8_01                      1           1           1             1             1
                  KLASA_MIEJSCOWOSCI WOJEWODZTWO Podregion_58 GF8_01
L_OSOB_2009                        1           1           1      1
L_OSOB_2011                        1           1           1      1
L.OSOB_2013                        0           0           0      0
WAGA_GD_2000                       1           1           1      1
WAGA_GD_2013                       0           0           0      0
KLASA_MIEJSCOWOSCI                 0           0           0      0
WOJEWODZTWO                        0           0           0      0
Podregion_58                       0           0           0      0
GF8_01                             1           1           1      0
Random generator seed value:  100
```

Najważniejsze argumenty funkcji:

- data – Ramka danych, zawierająca braki danych (brak danych = NA)
- m – liczba imputacji (domyślnie m = 5)
- method – wektor z nazwą zastosowanej metody imputacji (dla każdej zmiennej konieczne jest określenie metody)

- predictorMatrix – macierz, określająca, które zmienne mają być predyktorami danej cechy (domyślnie: macierz jedynek z zerami na głównej przekątnej)
- maxit – maksymalna liczba iteracji
- printFlag – wyświetla historię obliczeń (domyślnie printFlag = TRUE)
- seed – liczba całkowita, będąca argumentem generatora liczb pseudolosowych

Metoda imputacji różni się w zależności od typu zmiennej. Opcje argumentu *method* zostały wyróżnione w tabeli 4.

Tabela 4: Opcje argumentu *method* funkcji *mi* dla określonych typów zmiennych

Opcja argumentu *method*	**Opis**	**Typ zmiennej**
pmm	predykcyjne dopasowanie według średniej	numeryczna
norm	Bayesian linear regression	numeryczna
nob.norm	non-Bayesian linear regression	numeryczna
mean	imputacja średnią	numeryczna
2l.norm	dwupoziomowy model liniowy	numeryczna
logreg	regresja logistyczna	faktor (2 poziomy)
polyreg	Model logitowy	faktor (więcej niż 2 poziomy)
polr	uporządkowany model logitowy	porządkowa (więcej niż 2 poziomy)
sample	próba losowa z posiadanych danych	każdy typ

Źródło: opracowanie własne na podstawie [van Buuren and Groothuis-Oudshoorn 2011, s. 16].

Każdy z argumentów może byc używany w postaci odrębnej funkcji, np. *mice.impute.2l.norm*, *mice.impute.polr* itd. Ponadto, występują funkcje *mice.impute.logreg.boot* oraz *mice.impute.norm.boot*, które umożliwiają wykorzystanie bootstrapów.

Interesującym argumentem funkcji *mice* jest również *predictorMatrix*, który określa macierz, bedącą podstawą obliczeń. Macierz w zestawieniu 7 mówi nam, które zmienne w kolumnach są predyktorami wartości zmiennej w wierszu (przykładowo, zmienna L.OSOB_2013 nie zawiera braków danych, więc żadna ze zmiennych nie jest używana przy obliczaniu jej imputowanych wartości).

Szybki wybór predyktorów umożliwia funkcja *quickpred*, której wynik został przedstawiony w zestawieniu 8.

Pakiet *Amelia* W ramach tego pakietu najbardziej przydatna jest funkcja o tej samej nazwie, która posługuje się algorytmem EM i boostrapem przy zastosowaniu imputacji wielokrotnej. W przeciwieństwie do dwóch pakietów przedstawionych wcześniej, funkcja *Amelia* jest alternatywą dla metody MCMC [Yucel 2011].

Zestawienie 7: Macierz, będąca podstawą obliczeń przy pomocy funkcji *mice*

	L_OSOB_2009	L_OSOB_2011	L.OSOB_2013	WAGA_GD_2000	WAGA_GD_2013
L_OSOB_2009	0	1	1	1	1
L_OSOB_2011	1	0	1	1	1
L.OSOB_2013	0	0	0	0	0
WAGA_GD_2000	1	1	1	0	1
WAGA_GD_2013	0	0	0	0	0
KLASA_MIEJSCOWOSCI	0	0	0	0	0
WOJEWODZTWO	0	0	0	0	0
Podregion_58	0	0	0	0	0
GF8_01	1	1	1	1	1

	KLASA_MIEJSCOWOSCI	WOJEWODZTWO	Podregion_58	GF8_01
L_OSOB_2009	1	1	1	1
L_OSOB_2011	1	1	1	1
L.OSOB_2013	0	0	0	0
WAGA_GD_2000	1	1	1	1
WAGA_GD_2013	0	0	0	0
KLASA_MIEJSCOWOSCI	0	0	0	0
WOJEWODZTWO	0	0	0	0
Podregion_58	0	0	0	0
GF8_01	1	1	1	0

Zestawienie 8: Macierz wygenerowana za pomocą funkcji *quickpred*

	L_OSOB_2009	L_OSOB_2011	L.OSOB_2013	WAGA_GD_2000	WAGA_GD_2013
L_OSOB_2009	0	1	1	1	1
L_OSOB_2011	1	0	1	0	1
L.OSOB_2013	0	0	0	0	0
WAGA_GD_2000	1	0	1	0	0
WAGA_GD_2013	0	0	0	0	0
KLASA_MIEJSCOWOSCI	0	0	0	0	0
WOJEWODZTWO	0	0	0	0	0
Podregion_58	0	0	0	0	0
GF8_01	1	1	1	1	0

	KLASA_MIEJSCOWOSCI	WOJEWODZTWO	Podregion_58	GF8_01
L_OSOB_2009	1	1	1	1
L_OSOB_2011	1	1	1	1
L.OSOB_2013	0	0	0	0
WAGA_GD_2000	1	1	1	1
WAGA_GD_2013	0	0	0	0
KLASA_MIEJSCOWOSCI	0	0	0	0
WOJEWODZTWO	0	0	0	0
Podregion_58	0	0	0	0
GF8_01	1	1	1	0

Funkcja *amelia* pozwala określić liczbę imputowanych zbiorów danych (argument *m*) oraz numer kolumny, w której znajduje się zmienna typu ID (argument *idvars*). Zmienna tego typu zostaje wyłączona z obliczeń, ale dołączona do imputowanego zbioru danych w niezmienionej postaci. Pozostałe argumenty funkcji są opcjonalne i zostały opisane poniżej.

Podsumowanie zastosowania funkcji *amelia* umożliwia funkcja *summary*, której wynik zaprezentowano w zestawieniu 9. w podsumowaniu znajdują się informacje na temat:

- liczby imputowanych zbiorów danych
- liczby wierszy po usunięciu niekompletnych rekordów (*Rows after Listwise Deletion*)
- liczby zmiennych z pełnymi danymi (*Patterns of missingness in the data*)
- udziału braków danych w całkowitej liczbie obserwacji dla każdej zmiennej

Zestawienie 9: Podsumowanie zastosowania funkcji *amelia*

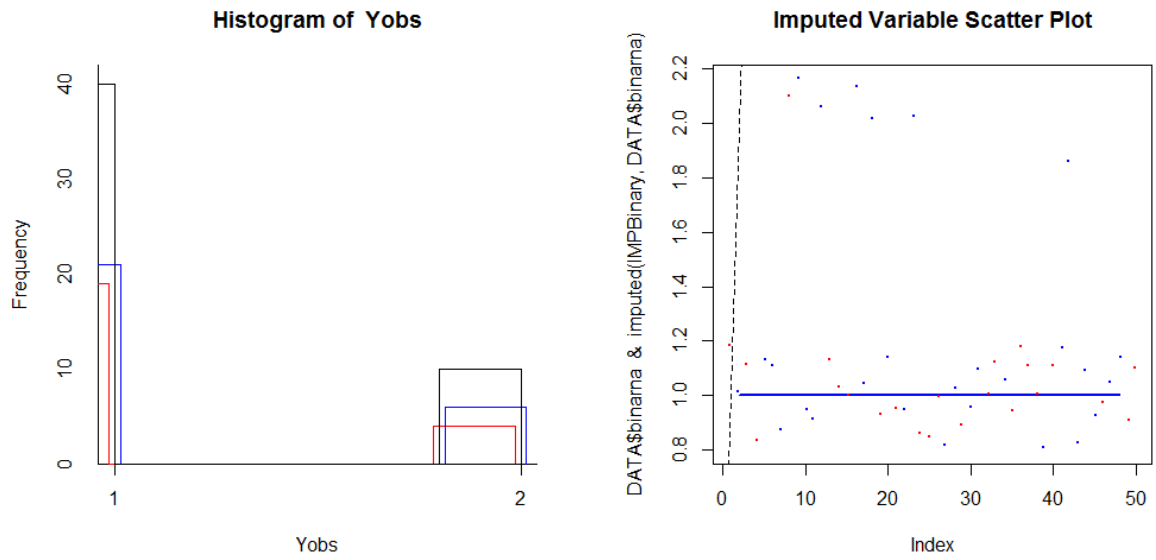
```
Amelia output with 3 imputed datasets.
Return code:  1
Message:  Normal EM convergence.

Chain Lengths:
--------------
Imputation 1:  389
Imputation 2:  79
Imputation 3:  199

Rows after Listwise Deletion:  8
Rows after Imputation:  50
Patterns of missingness in the data:  6

Fraction Missing for original variables:
-----------------------------------------

                    Fraction Missing
nr_sztywny_15062013             0.00
L_OSOB_2009                     0.54
L_OSOB_2011                     0.28
L.OSOB_2013                     0.00
WAGA_GD_2000                    0.28
WAGA_GD_2013                    0.00
KLASA_MIEJSCOWOSCI              0.00
WOJEWODZTWO                     0.00
Podregion_58                    0.00
GF8_01                          0.46
```

Funkcja *amelia* umożliwia imputację danych przekrojowych i szeregów czasowych[Yucel 2011]. W tym celu konieczna jest specyfikacja numerów kolumn, w których znajduje się określony typ szeregu. Przykłady argumentów [Honaker et al. 2011]:

- *ts* – dla szeregu czasowego
- *cs* – dane przekrojowe
- *polytime* – argument pozwalający określić stopień wielomianu w modelu imputacji, pozwalającego zniwelować efekt szeregu czasowego (wartości: 0 dla wartości stałych, 1 dla funkcji liniowej, 2 dla funkcji kwadratowej)
- *intercs* – zmienna logiczna, określająca, czy funkcja *polytime* ma różnić się w ramach danych przekrojowych

W pakiecie Amelia założono, że zmienne mają rozkład normalny [Honaker et al. 2011]. Dlatego wprowadzono argumenty, umożliwiające przekształcenia funkcji o rozkładzie różnym od normalnego. Przykładu argumentów, pozwalających dokonać transformacji danych:

- *logs* - wektor numerów kolumn lub nazw zmiennych, które wymagają przekształcenia zgodnie z modelem logliniowym
- *sqrts* - wektor numerów kolumn lub nazw zmiennych, które wymagają przekształcenia zgodnie z funkcją typu $y = \beta_i \sqrt{x}$



(a) Histogram – *mi.hist*

(b) Wykres punktowy – *mi.scatterplot*

Wykres 4: Wykresy wygenerowane za pomocą funkcji z pakietu *mi*

- *logstc* - wektor numerów kolumn lub nazw zmiennych, które wymagają przekształcenia zgodnie z modelem logistycznym dla zmiennych proporcjonalnych (charakterystyka zmiennej proporcjonalnej znajduje się powyżej)
- *noms* - wektor numerów kolumn lub nazw zmiennych nominalnych
- *ords* - wektor numerów kolumn lub nazw zmiennych porządkowych

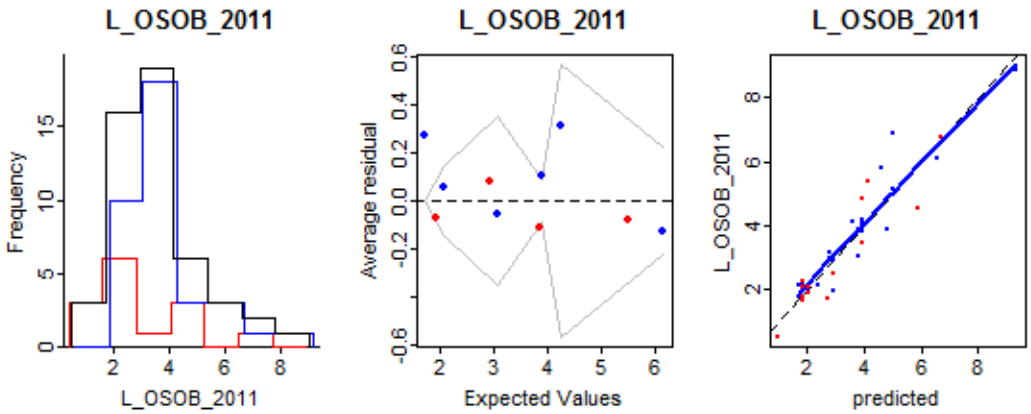
4 Miary służące ocenie wykorzystanych metod

4.1 Pakiet *mi*

Funkcja *mi.hist* Funkcja ta pozwala na graficzne przedstawienie częstości występowania wartości imputowanych w stosunku do rzeczywistych dla konkretnych realizacji zmiennej. Może być wykorzystywana po zastosowaniu funkcji odnoszących się do konkretnych typów zmiennej.

Na wykresie 4a kolorem czarnym zaznaczono histogram dla wszystkich wartości, czerwonym - dla wartości imputowanych i niebieskim - dla wartości rzeczywistych.

Funkcja *mi.scatterplot* Funkcja ta pozwala na stworzenie wykresu punktowego dla wartości rzeczywistych i imputowanych. Na wykresie 4b mputowane wartości oznaczone są kolorem czerwonym, a obserwowane - niebieskim. Jak widać, wartości imputowane nie odbiegają w zbyt dużym stopniu od wartości obserwowanych i położone są dość blisko linii trendu. Oznacza to, że model imputacji jest dobrze dopasowany.



Wykres 5: Wykres punktowy wygenerowany za pomocą funkcji *plot(miobject*

Funkcja *plot(mi.object)* Funkcja ta pozwala na stworzenie trzech wykresów dla wartości rzeczywistych i imputowanych danej zmiennej (jako przykład wybrano zmienną L_OSOB_2011) [Yu-Sung et al. 2011]. Na wykresie 5 wartości obserwowane zaznaczono niebieskim kolorem, imputowane - czerwonym, a ich sumę kolorem czarnym. Wygenerowane wykresy:

- **histogram** – interpretację wykresu tego typu przedstawiono przy omówieniu funckji *mi.hist*.
- **wykres punktowy średnich reszt** – przedstawia odległości między wartościami imputowanymi i rzeczywistymi oraz granice, w jakich powinny się mieścić wartości imputowane; na wykresie wartości imputowane nie wykraczają poza wyznaczone granice, co sugeruje, że model jest dobrze dopasowany.
- **wykres punktowy łączący wartości imputowane i obserwowane zmiennej z wartościami przewidywanymi** – na wykresie zaznaczono krzywą, która pozwala ocenić dopasowanie i modelu imputacji. Wartości imputowane są położone w pobliżu wartości obserwowanych i układają się wokół wyznaczonej linii, co oznacza, że model imputacji jest dobrze dopasowany.

4.2 Pakiet *mice*

Ocena za pomocą metod statystycznych W ramach pakietu *mice* dostępna jest funkcja *pool*, pozwalająca ocenić dopasowanie modelu imputacji. Przykładowy model zakłada, że zmienna L.OSOB_2013 jest powiązana ze zmiennymi L_OSOB_2009, i L_OSOB_2011 funkcją liniową. Wynik zastosowania omawianej funkcji przedstawiono w zestawieniu 10. Wartości p(tutaj $Pr(>|t|)$) wskazują, że jedynie zmienna L_OSOB_2011 ma istotny wpływ na zmienną L.OSOB_2013. Oznacza to, że tylko ta zmienna powinna być uwzględniona w modelu imputacji.

Porównanie dwóch modeli imputacji Pakiet *mice* umożliwia również porównanie dwóch modeli imputacji przy pomocy funkcji *pool.compare*. Pozwala ona zestawić ze sobą dwa mo-

Zestawienie 10: Wynik zastosowania funkcji *pool*

	est	se	t	df	Pr(>|t|)	lo 95	hi 95	nmis	fmi	lambda
(Intercept)	0.10	0.32	0.32	31.99	0.75	-0.55	0.75	NA	0.23	0.19
L_OSOB_2011	1.00	0.06	15.36	35.02	0.00	0.87	1.13	14	0.19	0.15
L_OSOB_2009	-0.01	0.10	-0.08	29.05	0.94	-0.22	0.20	27	0.28	0.23

dele, z których jeden uzależnia daną zmienną od większej liczby predyktorów niż drugi. Hipoteza zerowa wykorzystanego tu testu Walda mówi, że dodatkowe zmienne (te, które zostały uwzględnione w modelu z większą liczbą predyktorów) nie mają istotnego wpływu na zmienną, której wartości są imputowane [van Buuren and Groothuis-Oudshoorn 2011]. W tym wypadku porównano model przedstawiony w poprzednim akapicie z modelem, w którym uzależniono zmienną L.OSOB_2013 jedynie od zmiennej L_OSOB_2011 (również za pomocą funkcji liniowej). Otrzymano wartość p na poziomie 0.94, co oznacza, że zmienna L_OSOB_2009 nie powinna być wykorzystywana w modelu imputacji.

Funkcja *stripplot* pozwala na stworzenie wykresów punktowych dla wartości obserwowowanych (kolor jaśniejszy) i imputowanych (kolor ciemniejszy) dla poszczególnych wersji imputacji. Wykresy dla zmiennych, dla których dane są kompletne zawierają tylko kolor jasny (por. wykres 6). Na podstawie wykresu można zauważyć, że wartości imputowane nie odbiegają w duzym stopniu od obserwowanycg, więc model jest dopasowany dobrze.

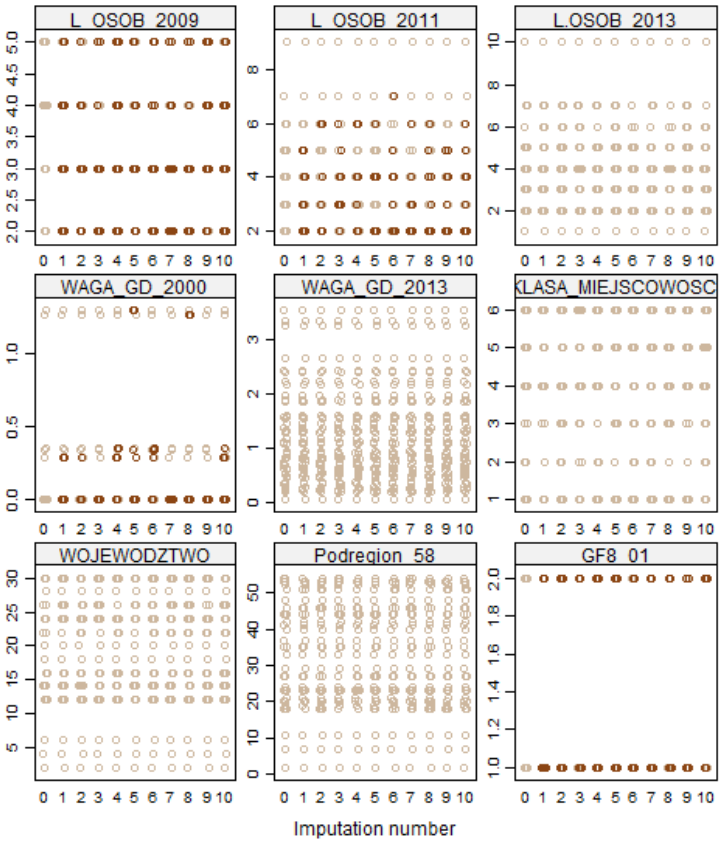
Funkcja *bwplot* pozwala na stworzenie wykresów pudełkowych, na podstawie których można porównać rozkład wartości obserwowanych i imputowanych. Na wykresie 7 kolorem jaśniejszym zaznaczono wykres dla danych rzeczywistych. Kolorem ciemniejszym zaznaczono wykresy dla poszczególnych wariantów imputowanych wartości danej zmiennej. Analiza różnic między poszczególnymi charakterystykami zmiennych (np. mediana) pozwala wybrać odpowiedni model imputacji.

Funkcja *densityplot* pozwala na wygenerowanie funkcji gęstości dla poszczególnych zestawów wartości imputowanych (kolor czerwony) i porównanie ich z funkcją gęstości dla wartości obserwowanych (kolor niebieski). Wynik zastosowania funkcji przedstawiono na wykresie 8.

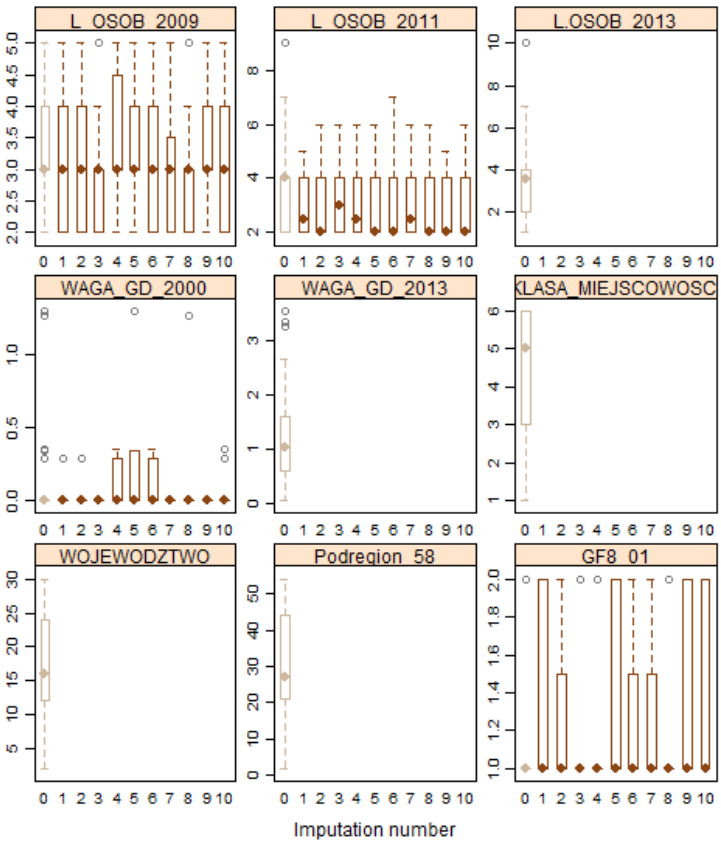
4.3 Pakiet *Amelia*

Funkcja *compare.density* pozwala wygenerować funkcje gęstości dla wartości imputowanych i obserwowanych. Na wykresie 9 przedstawiono zastosowanie funkcji dla zmiennej L_OSOB_2011.

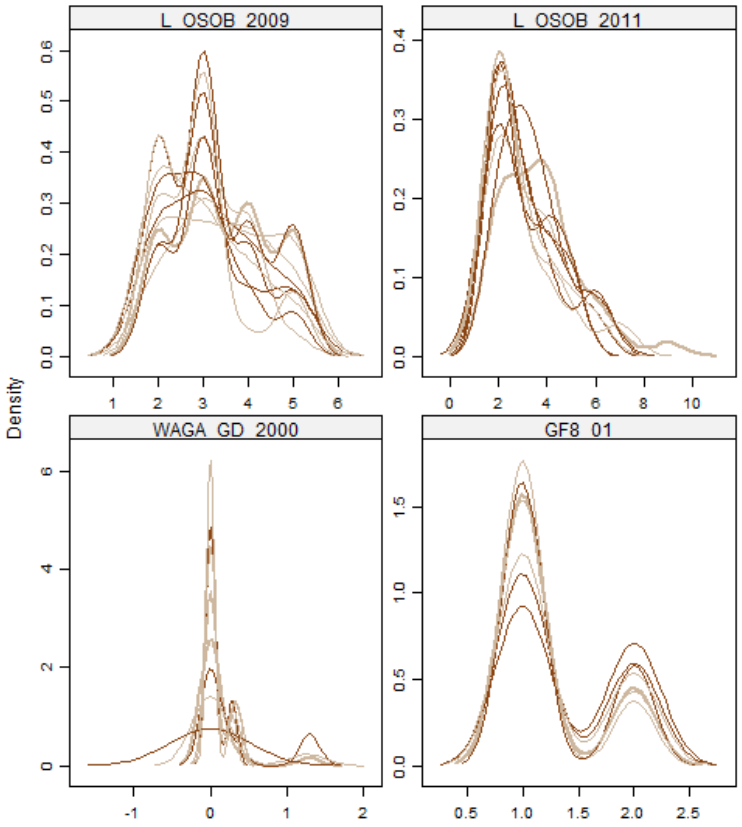
Funkcja (overimpute) generuje wykres punktowy dla przypisanych obserwowanym wartościom wartości imputowanych. Gdyby model imputacji był doskonały, na wykresie 10 wszystkie



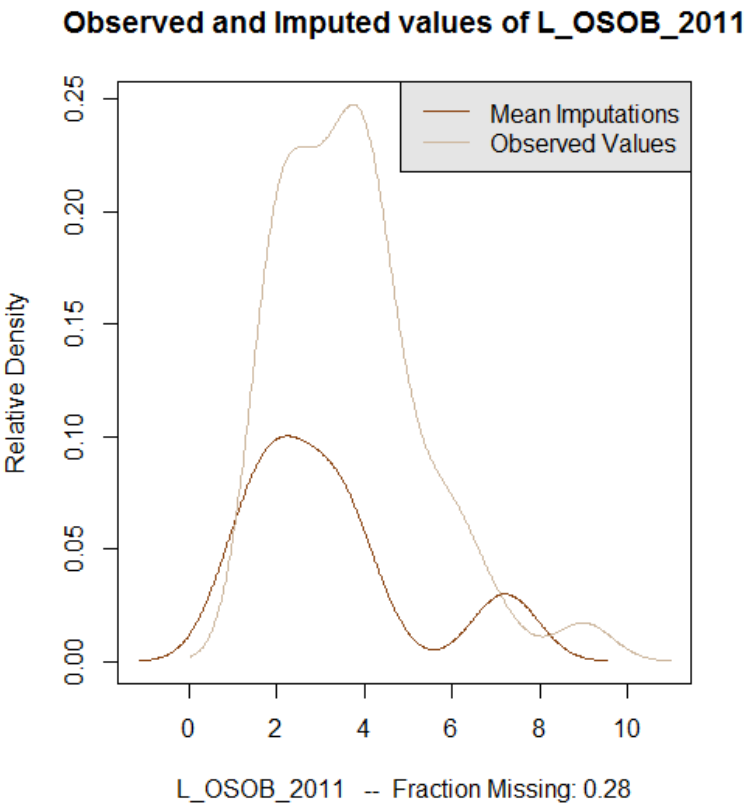
Wykres 6: Wykres punktowy wygenerowany za pomocą funkcji *stripplot*



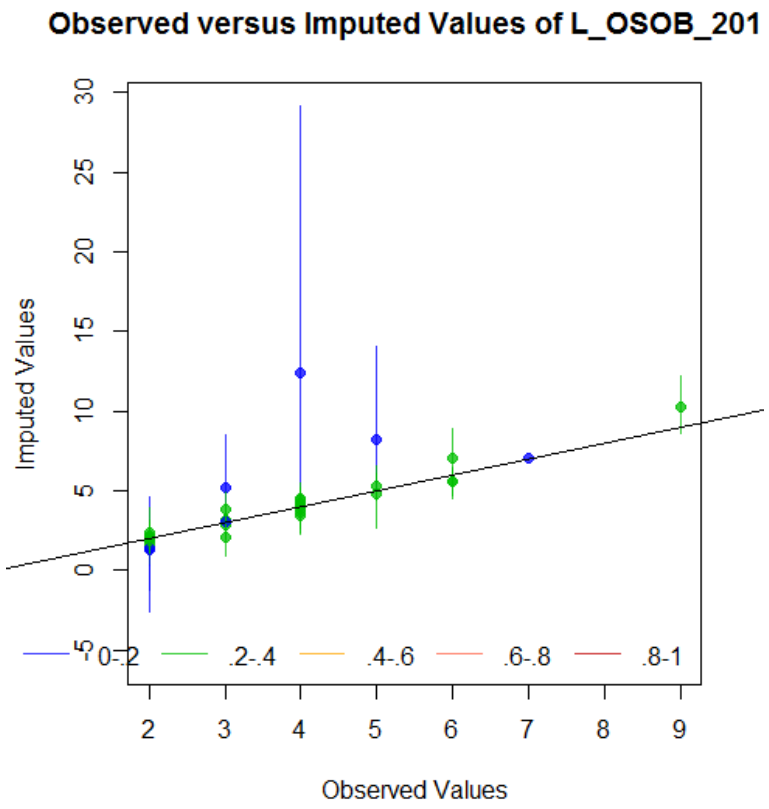
Wykres 7: Wykres punktowy wygenerowany za pomocą funkcji *bwplot*



Wykres 8: Wykres punktowy wygenerowany za pomocą funkcji *densityplot*



Wykres 9: Wykres punktowy wygenerowany za pomocą funkcji *compare.density*



Wykres 10: Wykres punktowy wygenerowany za pomocą funkcji *overimpute*

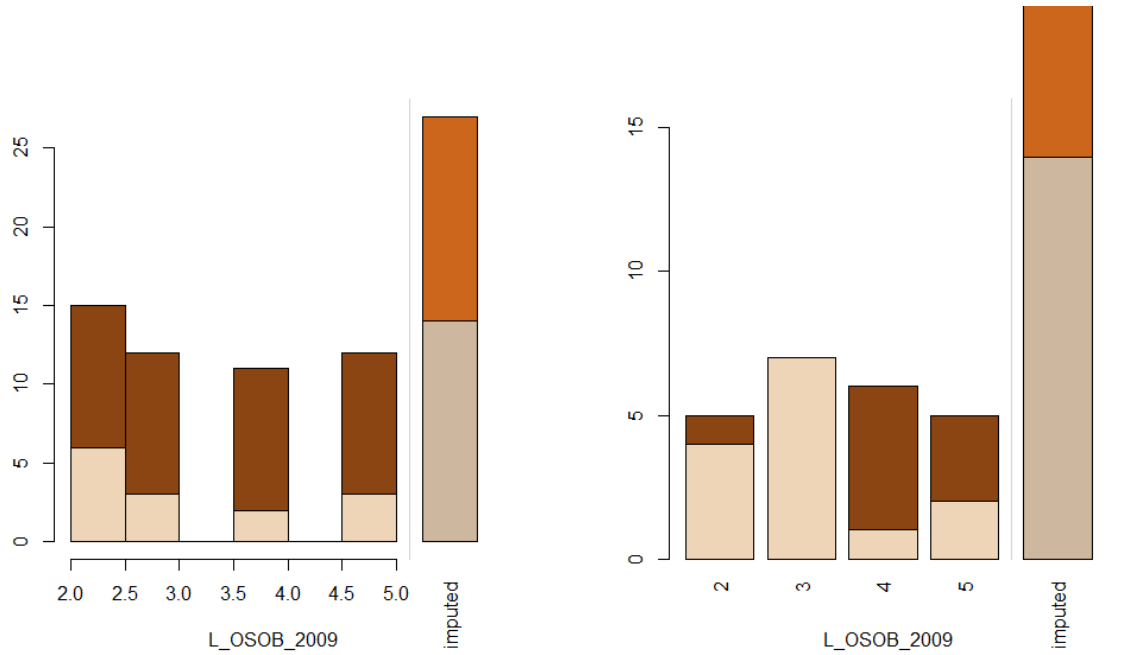
punkty danego koloru leżałyby na czarnej linii. Pionowe linie obrazują 90-procentowe przedziały ufności dla każdej wartości. Kolor tych linii pokazuje możliwy udział wartości brakujących we wzorze imputacji dla danej zmiennej [Honaker et al. 2011].

4.4 Porównanie zestawów imputacji w pakietach *mi* i *mice*

Porównanie wyników imputacji jest możliwe dzięki pakietowi VIM. Prostymi w zastosowaniu są funkcje *barMiss* oraz *pbox*.

Wynik zastosowania pierwszej funkcji został przedstawiony na wykresie 11. Jaśniejsze słupki na wykresie przedstawiają wartości obserwowane, a ciemniejsze - wartości brakujące/imputowane. Na osi pionowej umieszczono udział wartości danego typu.

Wynik zastosowania funkcji *pbox* przedstawiono na wykresie 12. Wykres biały odnosi się do wszystkich wartości, wykres ciemniejszy - do wartości brakujących, a wykres jaśniejszy - do wartości obserwowanych. Jak widać, różnice w wysokości mediany i pierwszego kwartyla są podobne na obydwu wykresach, jednak w zakresie różnic w wysokości trzeciego kwartyla można zauważyć istotne rozbieżności. Dla danych wygenerowanych przy pomocy pakietu mice wartość trzeciego kwartyla jest niższa. Wydaje się, że zestaw danych imputowanych, wygenerowany za pomocą funkcji *mi* jest bardziej wiarygodny, ponieważ zachowuje odpowiednie

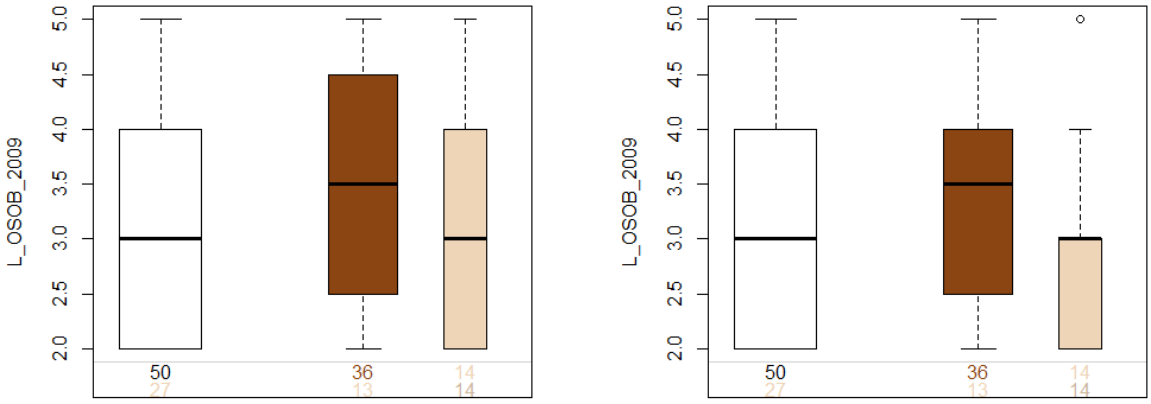


(a) Na podstawie zestawu danych z pakietu mi

(b) Na podstawie zestawu danych z pakietu mice

Wykres 11: Wykresy wygenerowane za pomocą funkcji *barMiss*

zróżnicowanie danych(zachowany jest odpowiedni poziom rozstępu międzykwartylowego).



(a) Na podstawie zestawu danych z pakietu mi

(b) Na podstawie zestawu danych z pakietu mice

Wykres 12: Wykresy wygenerowane za pomocą funkcji *pbox*

Literatura

James Honaker, Gary King, and Matthew Blackwell. Amelia ii: A program for missing data. *Journal of Statistical Software*, 45(7), 2011.

R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. 2002.

Roderick J. A. Little. Missing-data adjustments in large surveys. *Journal of Business and Economic Statistics*, 6(3):287–296, 1988.

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987.

J.L. Schafer. *Analysis of Incomplete Multivariate Data*. 1997.

Marcin Szymkowiak. *Estymatory kalibracyjne w badaniu budżetów gospodarstw domowych*. Uniwersytet Ekonomiczny w Poznaniu, 2009.

S. van Buuren. *Flexible Imputation of Missing Data*. 2012.

Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67, 2011.

Su Yu-Sung, Andrew Gelman, Jennifer Hill, and Masanao Yajima. Multiple imputation with diagnostics (mi) in r: Opening windows into the black box. *Journal of Statistical Software*, 45 (2), 2011.

Yang C. Yuan. Multiple imputation for missing data: Concepts and new development. *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, No. 267, 2000.

Recai M. Yucel. State of the multiple imputation software. *Journal of Statistical Software*, 45(1), 2011.

Spis treści

1 Metody imputacji i pakiety służące imputowaniu danych w R **1**

2 Funkcje wykorzystywane do podsumowań braków danych **3**

2.1 Tabelaryczne przedstawienie braków danych 3

2.2 Graficzne przedstawienie braków danych 5

3 Metody wykorzystywane w poszczególnych pakietach **8**

3.1 Imputacja wielokrotna 8

4 Miary służące ocenie wykorzystanych metod **17**

4.1 Pakiet *mi* 17

4.2 Pakiet *mice* 18

4.3 Pakiet *Amelia* 19

4.4 Porównanie zestawów imputacji w pakietach *mi* i *mice* 22

Spis tablic

1 Funkcje dostępne w pakietach wg metod imputacji 2

2 Funkcje służące podsumowaniu braków danych w ramach pakietu mice 3

3 Funkcja *mi* dla określonych typów zmiennych 12

4 Opcje argumentu *method* funkcji *mi* dla określonych typów zmiennych 14

Spis rysunków

1 Wykres wygenerowany za pomocą funkcji *missing.pattern.plot* 6

2 Wykres wygenerowany za pomocą funkcji *fluxplot* 7

3 Wykres wygenerowany za pomocą funkcji *missmap* 7

4 Wykresy wygenerowane za pomocą funkcji z pakietu *mi* 17

5 Wykres punktowy wygenerowany za pomocą funkcji *plot(miobject* 18

6 Wykres punktowy wygenerowany za pomocą funkcji *stripplot* 20

7 Wykres punktowy wygenerowany za pomocą funkcji *bwplot* 20

8 Wykres punktowy wygenerowany za pomocą funkcji *densityplot* 21

9 Wykres punktowy wygenerowany za pomocą funkcji *compare.density* 21

10 Wykres punktowy wygenerowany za pomocą funkcji *overimpute* 22

11 Wykresy wygenerowane za pomocą funkcji *barMiss* 23

12 Wykresy wygenerowane za pomocą funkcji *pbox* 24