

Pakiet mice

Martyna Majchrzak, Agata Makarewicz, Jacek Wiśniewski

26 03 2020



Wprowadzenie

MICE - Multivariate Imputation by Chained Equations
(wielowymiarowa imputacja za pomocą równań łańcuchowych)

Wykorzystanie

Pakiet mice zawiera funkcje służące do:

- ▶ generowania symulowanych niekompletnych danych (-> ampute)
- ▶ sprawdzenia wzorca brakujących danych (-> md.pattern, ...)
- ▶ imputacji brakujących danych (wielokrotnie) (-> mice)
- ▶ diagnozowania jakości imputowanych wartości (jakie funkcje?)
- ▶ analizy każdego uzupełnionego zbioru danych (?)
- ▶ zebrania wyników powtarzanych analiz (-> pool)
- ▶ przechowywania i eksportowania imputowanych danych w różnych formatach (?)
- ▶ (Incorporate custom imputation methods)?

Zbiory danych dostępne w pakiecie mice

- ▶ boys (wzrost, waga, wiek . . . duńskich chłopców)
- ▶ brandsma (dane o uczniach z różnych szkół)
- ▶ pattern1,2,3,4 (proste zbiory danych z różnymi wzorcami braków danych)

Generowanie braków danych

- ▶ funkcja `ampute`
- ▶ generowanie brakujących danych potrzebnych do symulacji
 - ▶ określony procent danych zostaje zastąpiony NA (obserwacje są wybierane losowo)
 - ▶ różne mechanizmy: MAR (Missing At Random), MCAR (Missing Completely At Random), MNAR (Missing Not At Random)
 - ▶ określenie wzorca braków danych oraz częstotliwości jego wystąpienia

```
set.seed(1)
```

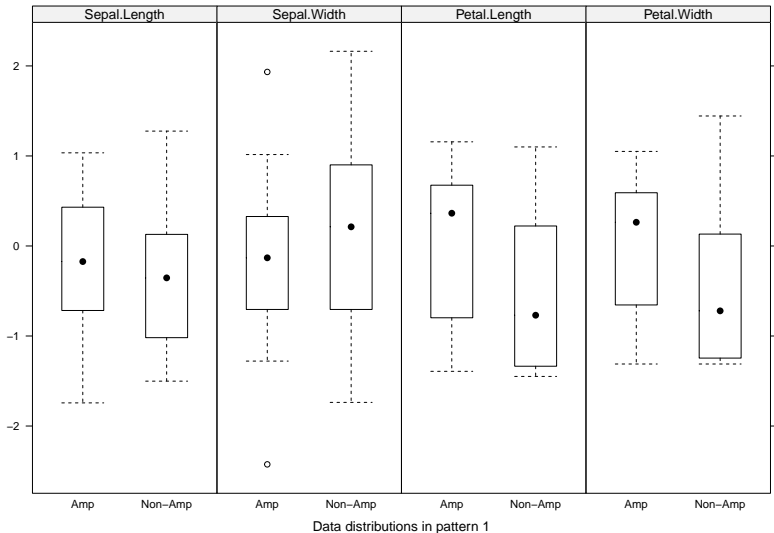
```
iris_amp <- ampute(iris[, -5], prop = 0.5, mech = "MCAR")
```

Sprawdzenie wzorca brakujących danych

Więkoszość metod do rysowania wykresów nadpisuje funkcje z pakietu `lattice`.

- ▶ `bwplot`
- ▶ `md.pattern`
- ▶ wyświetlenie wzorca brakujących danych w formie wykresu (oraz tabeli - w konsoli)
- ▶ `fluxplot <- # done`

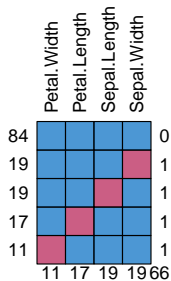
```
mice::bwplot(iris_amp, which.pat = 1)
```



```
md.pattern(iris_amp$amp, plot = TRUE, rotate.names = TRUE)
```



```
md.pattern(iris_amp$amp, plot = TRUE, rotate.names = TRUE)
```

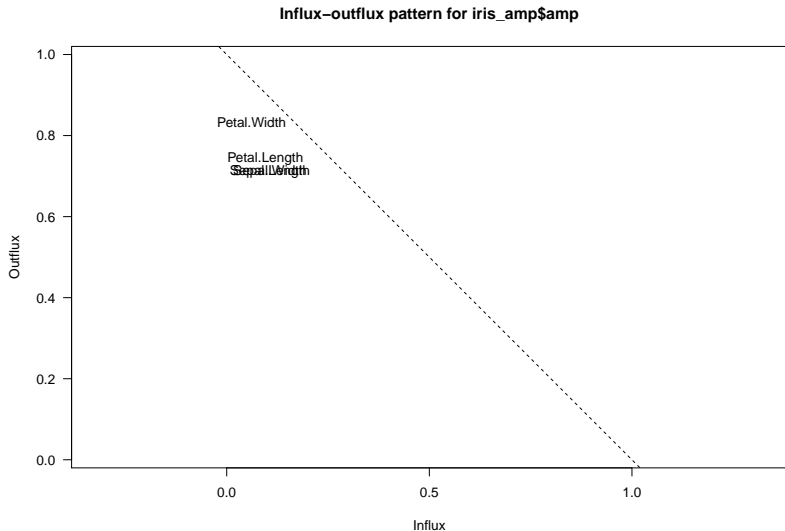


Fluxplot

Dla każdej zmiennej obliczane są 2 wartości:

- ▶ Influx - liczba par zmiennych takich, że w danej jest brak a w drugiej nie, podzielona przez wszystkie obserwacje. Dane pełne: 0, Dane całkowicie brakujące:1
- ▶ Outflux - liczba par zmiennych takich, że w danej jest obserwacja a w drugiej brak, podzielona przez wszystkie obserwacje. Dane pełne: 1, Dane całkowicie brakujące:0.
Potencjalna użyteczność do imputowania innych zmiennych.

```
fluxplot(iris_amp$amp)
```



Inne metody wizualizacji danych imputowanych

- ▶ `xyplot`
- ▶ `densityplot`
- ▶ `stripplot`

Imputacja danych

Zbiór danych boys

```
# zajmujemy sie boys bo maja ordered/unordered factor - n  
str(boys)
```

```
## 'data.frame':    748 obs. of  9 variables:  
## $ age: num  0.035 0.038 0.057 0.06 0.062 0.068 0.068 0  
## $ hgt: num  50.1 53.5 50 54.5 57.5 55.5 52.5 53 55.1 54  
## $ wgt: num  3.65 3.37 3.14 4.27 5.03 ...  
## $ bmi: num  14.5 11.8 12.6 14.4 15.2 ...  
## $ hc : num  33.7 35 35.2 36.7 37.3 37 34.9 35.8 36.8 38  
## $ gen: Ord.factor w/ 5 levels "G1"<"G2"<"G3"<...: NA NA  
## $ phb: Ord.factor w/ 6 levels "P1"<"P2"<"P3"<...: NA NA  
## $ tv : int   NA NA NA NA NA NA NA NA NA NA NA ...  
## $ reg: Factor w/ 5 levels "north","east",...: 4 4 4 4 4
```

Zbiór zawiera już braki danych, ma kolumny: - numeryczne -
kategoryczne uporządkowane - kategoryczne nieuporządkowane

Funkcja mice

W zależności od typu brakujących danych, funkcja mice przyjmuje jako parametr inne metody imputacji danych.

Dane podzielone są na 4 kategorie:

- ▶ dane numeryczne (ciągłe)
- ▶ dane binarne (dane typu factor z dwoma poziomami)
- ▶ nieuporządkowane dane katégoryczne (dane typu factor z więcej niż 2 poziomami)
- ▶ uporządkowane dane katégoryczne (dane typu factor z więcej niż 2 poziomami uporządkowanymi)

Dowolne dane

Niektóre metody imputacji możemy zastosować do każdego typu danych.

- ▶ pmm (predictive mean matching/ predykcyjne dopasowanie średniej)
- ▶ midastouch (weighted predictive mean matching/ ?)
- ▶ sample (losowa próbka)
- ▶ cart (drzewo klasyfikacyjne i regresji (?))
- ▶ rf (random forest/lasy losowe)
- ▶ 2lonly.pmm (Level-2 class predictive mean matching) <- ?

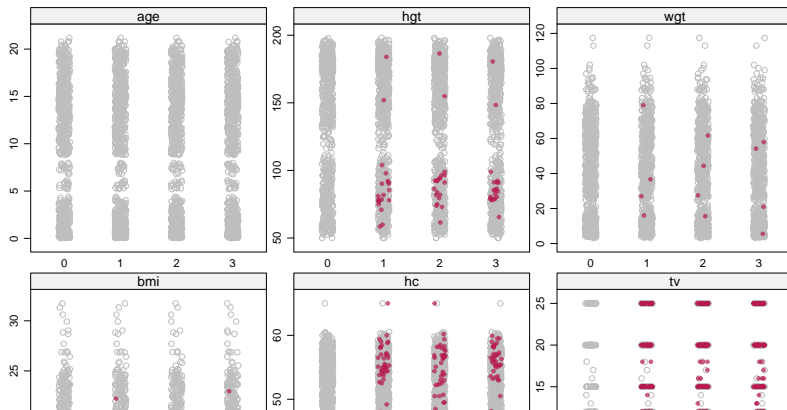
Dane numeryczne

- ▶ pmm (domyślna)
- ▶ mean (średnia)
- ▶ norm (Bayesian linear regression/regresja liniowa)
 - ▶ norm.nob (linear regression ignoring model error)
 - ▶ norm.boot (linear regression using bootstrap)
 - ▶ norm.predict (linear regression, predicted values)
- ▶ quadratic (imputation of quadratic terms)
- ▶ ri (random indicator for nonignorable data)

```
dutch_boys<-boys  
imp <- mice(dutch_boys[,-c(6,7,9)],  
            method="pmm", m=3, maxit=3)
```

```
dutch_boys[,-c(6,7,9)] <- complete(imp)
```

```
stripplot(imp,col=c("grey",mdc(2)),pch=c(1,20))
```

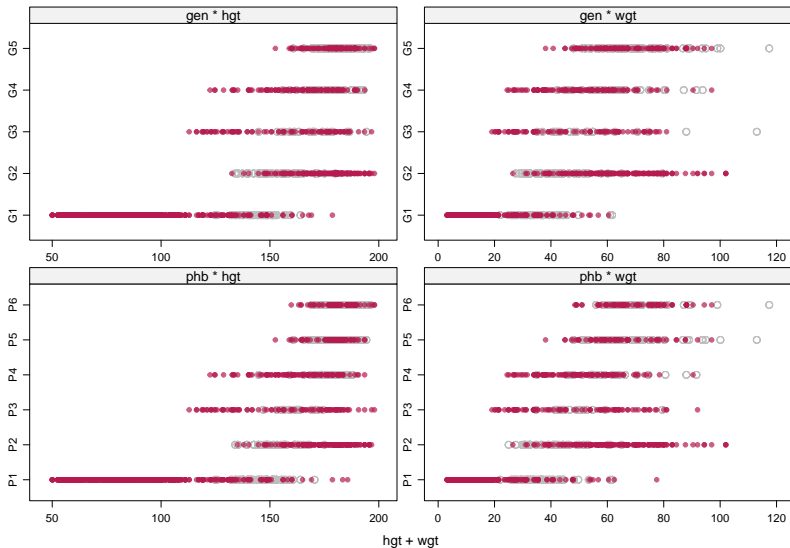


Nieuporządkowane dane kategoryczne

- ▶ polyreg (Polytomous logistic regression) (domyślna)
- ▶ lda (liniowa analiza dyskryminacyjna)

```
imp <- mice(dutch_boys[, -9], method="lda", m=3, maxit=3)  
dutch_boys[, -9] <- complete(imp)
```

```
xyplot(imp, gen+phb ~ hgt+wgt,  
       cex=1,col=c("grey",mdc(2)),pch=c(1,20))
```

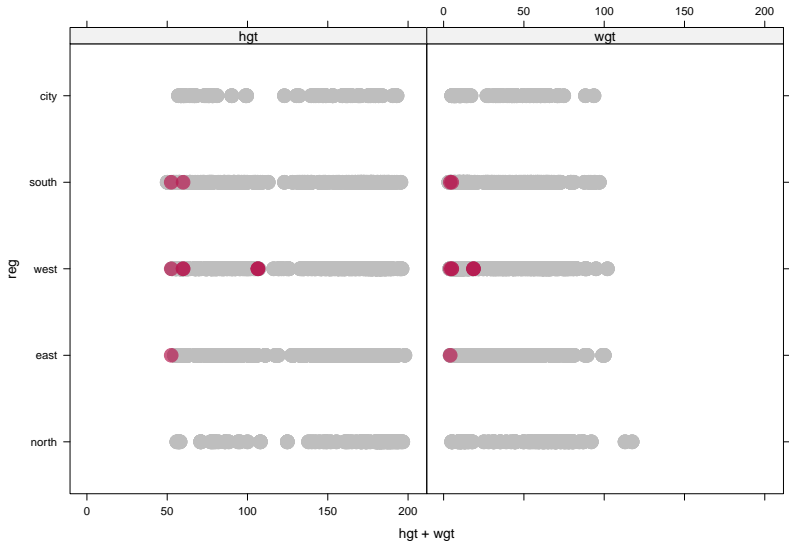


Uporządkowane dane kategoryczne

- ▶ polr (Proportional odds model) (domyślna)

```
imp <- mice(dutch_boys, method="polr", m=3, maxit=3)  
dutch_boys <- complete(imp)
```

```
xyplot(imp,reg ~ hgt+wgt,  
       cex=3,col=c("grey",mdc(2)),pch=20)
```



Dane binarne

- ▶ logreg (logistic regression/regresja logistyczna) (domyślna)
- ▶ logreg.boot (logistic regression with bootstrap)


```
# Ponieważ w zbiorze danych boys nie ma danych binarnych,  
# posłużymy się zbiorem mtcars.
```

```
# Usuwamy tylko w tych dwóch kolumnach: vs i am
```

```
mtcars_amp<-ampute(data=mtcars,  
                    patterns=rbind(  
                        c(1,1,1,1,1,1,1,0,1,1,1),  
                        c(1,1,1,1,1,1,1,1,0,1,1)),  
                    prop = 0.5,  
                    mech="MCAR")$amp  
mtcars_amp[,8] <- as.factor(mtcars_amp[,8])  
mtcars_amp[,9] <- as.factor(mtcars_amp[,9])  
str(mtcars_amp)
```

```
## 'data.frame':    32 obs. of  11 variables:
```

```
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 1
```

```
## $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
```

```
## $ disp: num  160 160 108 258 360 ...
```

```
## $ hp : num  110 110 93 110 175 105 245 62 95 123 ...
```

```
## $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92
```

```
## $ wt : num  2.62 2.88 2.32 3.21 3.44
```

```
imp <- mice(mtcars_amp[,c(8,9)], method="logreg", m = 3, ma
```

```
mtcars_amp[,c(8,9)] <- complete(imp)  
str(mtcars_amp)
```

```
## 'data.frame':    32 obs. of  11 variables:  
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 1  
## $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...  
## $ disp: num  160 160 108 258 360 ...  
## $ hp : num  110 110 93 110 175 105 245 62 95 123 ...  
## $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92  
## $ wt : num  2.62 2.88 2.32 3.21 3.44 ...  
## $ qsec: num  16.5 17 18.6 19.4 17 ...  
## $ vs : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 2 1 2  
## $ am : Factor w/ 2 levels "0","1": 2 2 2 1 1 2 2 2 1 1  
## $ gear: num  4 4 4 3 3 3 3 4 4 4 ...  
## $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Zebranie wyników analiz

► Funkcja pool

```
temp <- mice(dutch_boys, m = 20, maxit = 5, seed = 123)
modelFit <- with(temp, lm(age ~ hgt + wgt))
```

```
summary(pool(modelFit))
```

##	term	estimate	std.error	statistic	df
## 1	(Intercept)	-7.41446991	0.239298531	-30.98418	742.9334
## 2	hgt	0.10572022	0.003260907	32.42049	742.9334
## 3	wgt	0.07320633	0.005841462	12.53219	742.9334